P802.11F/D6, March 2003
DS5915

P802.11F
Draft Recommended Practice for Multi-Vendor Access Point Interoperability via an Inter-Access Point Protocol Across Distribution Systems Supporting IEEE 802.11 Operation

Prepared by the LAN MAN Standards Committee
of the
IEEE Computer Society

IEEE Standards Activities Department
Standards Licensing and Contracts
445 Hoes Lane, P.O. Box 1331
Piscataway, NJ 08855-1331, USA

Draft
Recommended Practice for Multi-Vendor Access Point Interoperability via an Inter-Access Point Protocol Across Distribution Systems Supporting IEEE 802.11 Operation

Sponsored by the
LAN/MAN Standards Committee
of the
IEEE Computer Society

IEEE Standards Department
Standards Licensing and Contracts
445 Hoes Lane, P.O. Box 1331
Piscataway, NJ 08855-1331, USA

Introduction

(This introduction is not part of IEEE P802.11f, Recommended Practice for Multi-Vendor Access Point Interoperability via Inter-Access Point Protocol Across Distribution Systems Supporting IEEE 802.11 Operation.)

At the time this standard was completed, the working group had the following membership:

Stuart Kerry, *Chair*
David Bagby, *Chair, Task Group f*
Bob O'Hara, *Editor, Task Group f*

Put working group member names here

The following persons were on the balloting committee: (To be provided by IEEE editor at time of publication.)

_____

Contents

Figures

Tables

Draft
Recommended Practice for Multi-Vendor Access Point Interoperability via an Inter-Access Point Protocol Across Distribution Systems Supporting IEEE 802.11 Operation

1 Overview

1.1 Scope

The scope of this document is to describe recommended practices for implementation of an Inter-Access Point Protocol (IAPP) on a Distribution System (DS) supporting ISO/IEC 8802-11:1999, IEEE Standard 802.11, wireless LAN (WLAN) links. The recommended DS utilizes an Inter-Access Point Protocol that provides the necessary capabilities to achieve multi-vendor Access Point (AP) interoperability within the DS. This IAPP is described for a DS consisting of IEEE 802 LAN components utilizing an Internet Engineering Task Force (IETF) Internet Protocol (IP) environment. Throughout this recommended practice, the terms ISO/IEC 8802-11:1999, IEEE 802.11, 802.11, and IEEE Std. 802.11-1999 are used interchangeably to refer to the same document, ISO/IEC 8802-11:1999 and its amendments and supplements published at the time this recommended practice was adopted.

1.2 Purpose

IEEE 802.11 specifies the MAC and PHY layers of a WLAN system and includes the basic architecture of such systems, including the concepts of APs and DSs. Implementations of these concepts were purposely not defined by 802.11 because there are many ways to create a WLAN system. Additionally, many of the possible implementation approaches involve higher network layers. While this leaves great flexibility in DS and AP functional design, the associated cost is that physical AP devices are unlikely to interoperate across a DS. In particular, the enforcement of the restriction that a station (STA) has a single association at a given time is unlikely to be achieved.

As 802.11 systems have grown in popularity, it has become clear that there are a small number of DS environments that comprise the bulk of the commercial and private WLAN system installations.

This recommended practice specifies the information to be exchanged between APs amongst themselves and higher layer management entities to support the 802.11 DS functions. The information exchanges are specified for DSs built on the IETF IP in a manner sufficient to enable the interoperation of DSs containing APs from different vendors that adhere to the recommended practice.

1.3 Inter-AP recommended practice overview

This recommended practice describes a service access point (SAP), service primitives, a set of functions and a protocol that will allow APs to interoperate on a common DS, using the Transmission Control Protocol over IP (TCP/IP) or User Datagram Protocol over IP (UDP/IP) to carry IAPP packets between APs, as well as describing the use of the Remote Authentication Dial-in User Service (RADIUS) Protocol, so APs may obtain information about one another. A proactive caching mechanism is also described that provides faster roaming times by sending STA context to neighboring APs. The devices in a network

that might use the IAPP are 802.11 APs. Other devices in a network that are affected by the operation of the IAPP are layer 2 networking devices, such as bridges and switches.

Throughout this recommended practice, reference is made to an "AP management entity" (APME). These are references to a function that is external to the IAPP, though likely still a function of the AP device. Typically, this management entity is the main operational program of the AP, implementing an AP manufacturer's proprietary features and algorithms, and incorporating the station management entity (SME) of 802.11. Figure 1 depicts an architecture of a typical AP in which the IAPP operates. The grey areas indicate areas where there is an absence of connection between blocks. The IAPP services are accessed by the APME through the IAPP SAP. The IAPP SAP is shown in Figure 1, as the small block between the APME and the IAPP blocks. IAPP service primitives are defined that allow the AP management entity to cause the IAPP to perform some function or to communicate with other APs in the DS or with a RADIUS server. Other service primitives indicate to the AP management entity that operations have taken place at other APs in the DS that can have an effect on information local to the AP.

The invocation of some IAPP service primitives relies on the RADIUS protocol to implement certain functions that are required for the correct and secure operation of the IAPP. In particular, the IAPP entity must be able to find and use a RADIUS server to look up the IP addresses of other APs in the ESS when given the BSSIDs of those other APs (if a local capability to perform such a translation is not present), and to obtain security information to protect the content of certain IAPP packets. The RADIUS server must provide extensions for IAPP-specific operations. There is currently work being performed in the IETF on RADIUS client configuration. Refer to the Internet Draft written by Robert Moskowitz on the topic or "RADIUS client kickstart". This work may be useful in conjunction with this recommended practice.

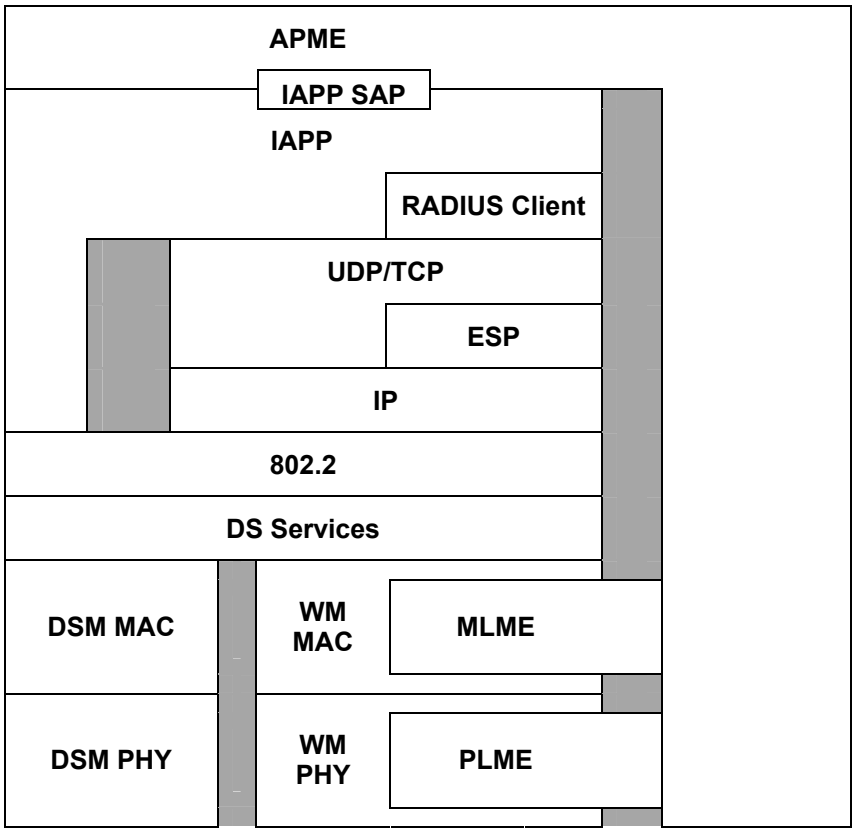

Figure 1 - AP Architecture with IAPP

The IAPP is not a routing protocol. The IAPP does not deal directly with the delivery of 802.11 data frames to the STA; instead the DS utilizes existing network functionality for data frame delivery. The data delivery service of the DS will function as desired when the STAs maintain a network layer address, e.g., IP address, or addresses that are valid for their point

of connection to the network, i.e., when a STA associates or reassociates, the STA must ascertain that its network layer address(es) is configured such that the normal routing functions of the network attaching to the BSS will correctly deliver the STA's traffic to the BSS to which it is associated. If the mobile device incorporating the STA determines that the network layer address(es) is not configured so as to allow the normal routing functions of the network to deliver the STA's traffic to the BSS to which it is associated, the STA must obtain such an address(es), before any network traffic can be delivered to it. A STA can meet the local IP address requirement in many ways. Two mechanisms for a STA to accomplish this are to renew a Dynamic Host Configuration Protocol (DHCP) lease for its IP address and to use Mobile IP to obtain a local IP address. Other mechanisms are possible that meet this requirement.

With the requirement that STAs maintain a valid network layer address, APs function much the same as 802.1D bridges. Additionally, the IAPP supports the following functions:

- DS Services, as defined in ISO/IEC 8802-11:1999
- Address mapping of wireless medium addresses of APs (their BSSID) to DS network layer addresses (IP addresses)
- Formation of a DS
- Maintenance of the DS
- Enforcement of the restriction of ISO/IEC 8802-11:1999 that a STA may have only a single association at any given time
- Transfer of STA context information between APs

IAPP transactions are over the DS. Hence, IAPP is independent of the security scheme defined in ISO/IEC 8802-11:1999.

This recommended practice makes use of the IETF RFCs listed in clause 2 to implement many of its functions. It also relies on a STA making use of the 802.11 Reassociation Request frame when roaming from one AP to another, in order to provide the most complete services to the APs using the IAPP. When a STA uses the 802.11 Association Request, rather than the Reassociation Request, the IAPP may not be able to notify the AP at which the STA was previously associated of the new association. This may result in the old AP (indicated in the "current AP" field of the reassociation request frame) maintaining context for the STA that has roamed to a new AP for a longer time than is strictly necessary. This may cause undue waste of resources at the old AP, as well as limiting the ability of the IAPP to help enforce the single STA association requirement of 802.11.

One issue that AP designers should address that can cause significant problems in a WLAN is the continued operation of an AP when it has lost its link to the DS. When an AP continues to accept associations without a link to the DS, it is a black hole in the WLAN, where STAs associate and cannot communicate with anything beyond the AP's BSS. When an AP loses its link to the DS, it should cease transmitting Beacons, disassociate all associated stations, and cease responding to Probe Request, Authentication, and Association Request frames.

1.4 Inter-AP Security Risks

Inter-AP communications present opportunities to an attacker. The attacker can use IAPP or forged 802.11 MAC management frames as a Denial-of-Service (DoS) attack against a STA state in its AP. It can capture MOVE packets to gather information on the STA that is roaming. It can act as a rogue AP in the ESS.

A bogus MOVE or ADD-Notify might cause an AP to drop all state it has with a STA. Since these IAPP packets are transmitted over IP, they could be introduced anywhere, from any device that has the necessary knowledge. This attack can best be eliminated by providing packet authentication to all MOVE and ADDs. The protection for the MOVEs can be provided by ESP (RFC 2406) pair-wise Security Associations (SA). The protection for the ADDs requires a group ESP Security Association. The content of the MOVE can be encrypted by the same ESP pair-wise SAs, protecting it from scrutiny of an attacker.

The use of ESP with RADIUS for the Key Management provides for discovery of Rogue APs. The use of ESP for IAPP MOVEs prevents a STA from roaming from a Rogue AP to a valid AP in the ESS. It also blocks the move of the STA context information to a Rogue AP if the STA roams to it. The RADIUS Access-Request provides the RADIUS server with knowledge of the presence of a Rogue AP.

2 References

The following standards contain provisions which, through references in this text, constitute provisions of this recommended practice. At the time of publication, the editions indicated were valid. All standards are subject to revision. When a standard is superceded by an approved revision, the revision applies.

IEEE Standard 802.11-1999[1]
IEEE Standard 802.1X-2001 Port Based Network Access Control[1]
IEEE Standard 802.2-1998 Logical Link Control[1]
RFC 768 – User Datagram Protocol[2]
RFC 791 – Internet Protocol[2]
RFC 1112 - Host extensions for IP multicasting[2]
RFC 1305 – Network Time Protocol version 3 specification[2]
RFC 1812 – Requirements for IP version 4 Routers[2]
RFC 2002 - IP Mobility Support[2]
RFC 2131 – Dynamic Host Configuration Protocol[2]
RFC 2181 - Clarifications to the DNS Specification[2]
RFC 2390 – Inverse Address Resolution Protocol[2]
RFC 2406 - IP Encapsulating Security Payload (ESP)[2]
RFC 2407 - The Internet IP Security Domain of Interpretation for ISAKMP[2]
RFC 2411 – IP Security Document Roadmap[2]
RFC 2548 - Microsoft Vendor-specific RADIUS Attributes[2]
RFC 2857 - The Use of HMAC-RIPEMD-160-96 within ESP and AH[2]
RFC 2865 - Remote Authentication Dial In User Service (RADIUS)[2]
RFC 2869 - RADIUS Extensions[2]
RFC 3162 – RADIUS in IPv6[2]
draft-moskowitz-radius-client-kickstart-00.txt – RADIUS Client Kickstart[2]

[1] IEEE publications are available from the Institute of Electrical and Electronics Engineers, 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ 08855-1331, USA (http://www.standards.ieee.org/).
[2] Requests for Comments (RFCs) are available from the Internet Engineering Task Force (IETF) (www.ietf.org)

3 Definitions, abbreviations, and acronyms

AAA	Authentication, Authorization, and Accounting
AP	Access Point
APME	Access Point Management Entity
BSS	Basic Service Set
BSSID	Basic Service Set Identifier
DHCP	Dynamic Host Configuration Protocol
DS	Distribution System
DSM	Distribution System Medium
ESP	IP Encapsulating Security Payload
ESS	Extended Service Set
IANA	Internet Assigned Numbers Authority
IAPP	Inter-Access Point Protocol
IETF	Internet Engineering Task Force
IP	Internet Protocol
LLC	Logical Link Control
MAC	Medium Access Control
MLME	MAC Layer Management Entity
PAE	Port Access Entity
PHY	Physical layer
PLME	PHY Layer Management Entity
RADIUS	Remote Authentication Dial In User Service
SA	Security Association
SAP	Service Access Point
SME	Station Management Entity
SPI	Security Parameter Index
SSID	Service Set Identifier
STA	Station
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
URL	Uniform Resource Locator
VSA	Vendor-specific attribute
WM	Wireless Medium
XID	Exchange Identifier

4 IAPP Service definition

The IAPP entity provides services to an AP in which it resides through the IAPP SAP. The SAP allows the management entity of the AP (APME) to invoke IAPP services and receive indications of service invocations at other APs in a single ESS. This clause defines the services that are available at the SAP. There are four types of service primitives that exist at the SAP. They are requests, confirms, indications, and responses. Service requests and responses are submitted to the IAPP entity by the entity at the next higher layer. In this document, the next higher layer is the APME. Service confirms and indications are delivered by the IAPP entity to the entity at the next higher layer.

This clause provides an abstract description of the services that an implementation should provide in order to interoperate with other implementations of the IAPP. This is not an exposed interface. A diagram of the relationships between the primitives is shown in Figure 2.

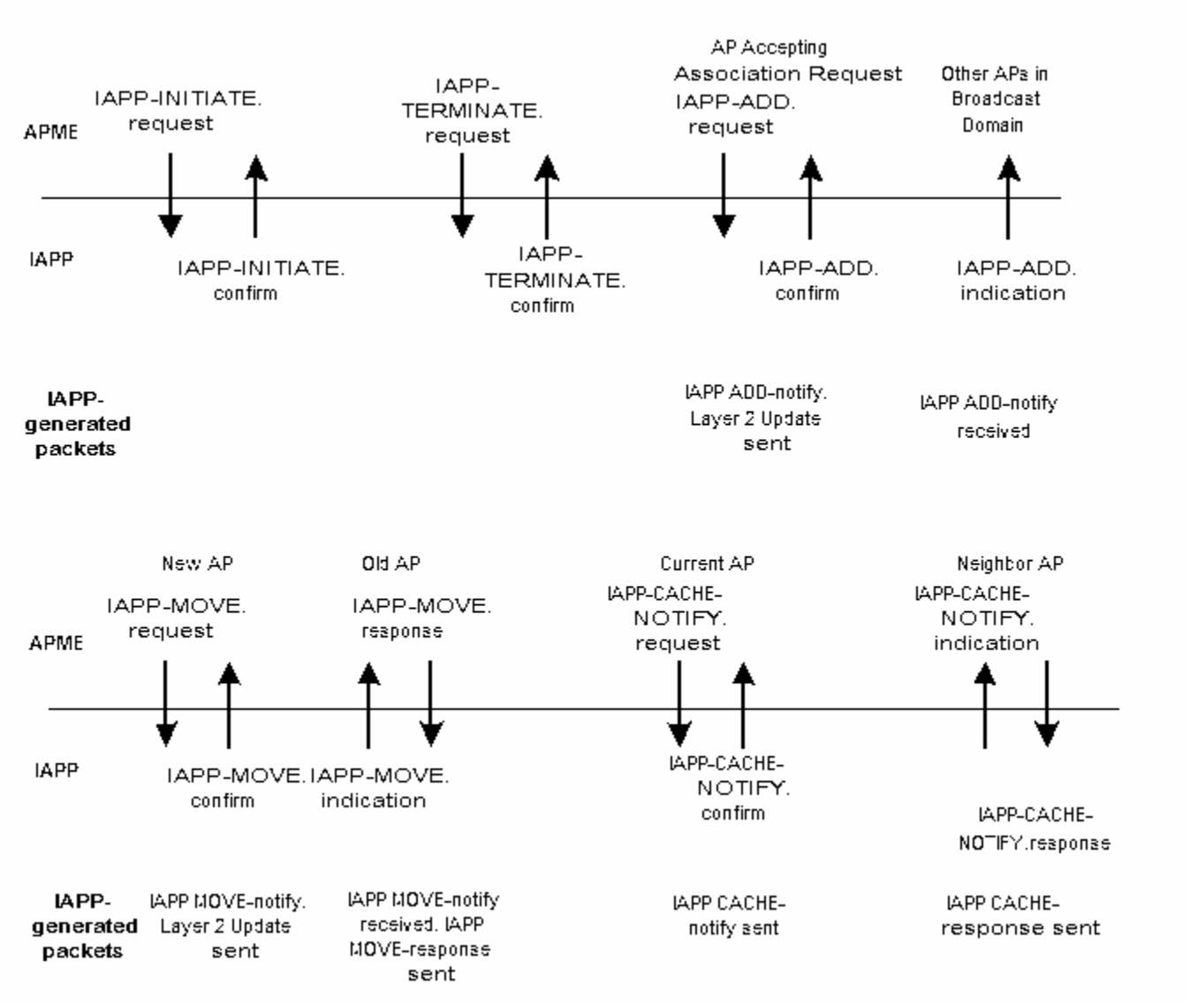

Figure 2 - Primitive Relationships

4.1 IAPP-INITIATE.request

4.1.1 Function

This service primitive causes the AP to initialize the IAPP entity, including its data structures, functions, and protocol.

4.1.2 Semantics of the service primitive

The IAPP-INITIATE.request has the following semantics.

```
IAPP-INITIATE.request {
                        TCP Port,
                        UDP Port,
                        IP Address,
                        BSSID Secret
                        }
```

The UDP Port parameter is the UDP port number to be opened for the IAPP for transmission and reception of IAPP packets. The port assigned by IANA for this protocol is 3517.

The TCP Port parameter is the TCP port number that the IAPP entity opens to listen for new IAPP TCP connections from the IAPP entities of other APs. The port assigned by IANA for this protocol is 3517.

The IP address is the IP address or fully qualified domain name of the RADIUS server.

The BSSID Secret is used to provide integrity, authentication, and confidentiality of the security block sent between the RADIUS server and the AP. See 5.3. If RADIUS is not used, this parameter is null.

4.1.3 When generated

This service primitive is generated by an APME to initiate the operation of the IAPP. At the time the IAPP-INITIATE.request is generated, the BSS controlled by this AP should not be operating, and no STAs should be associated with this AP. If necessary, the APME can issue an 802.11 MLME-RESET.request prior to generation of the IAPP-INITIATE.request.

4.1.4 Effect of receipt

Upon receipt of this service primitive from an APME, the action of the IAPP entity depends on the service level that is implemented. See 5.2. For level 1, there is no RADIUS usage. For levels 2 and 3, the IAPP entity sends the RADIUS Initiate-Request and receives the RADIUS Initiate-Accept or Initiate-Reject. If the Initiate-Accept is received, then the IAPP entity initializes its data structures, functions, and protocols. The port for the IAPP should be opened by the IAPP entity at this time. The previous information in any IAPP data structures is lost. If an Initiate-Reject is received, the IAPP does not start.

4.2 IAPP-INITIATE.confirm

4.2.1 Function

This service primitive notifies an APME that the actions begun by an IAPP-INITIATE.request have been completed.

4.2.2 Semantics of the service primitive

The IAPP-INITIATE.confirm primitive has the following semantics.

```
IAPP-INITIATE.confirm {
                        Status
                        }
```

The Status parameter indicates the result of the corresponding IAPP-INITIATE.request. The allowable value for the Status parameter are SUCCESSFUL, RUNNING, and FAILURE. SUCCESSFUL status should be returned if the IAPP entity is able to complete its initialization and open the requested port for the IAPP. RUNNING status should be returned if the IAPP entity receives an IAPP-INITIATE.request when the entity is already running. When RUNNING status has been returned, the IAPP

ignored the parameters from the corresponding IAPP-INITIATE.request and the operation of the IAPP was unaffected. FAILURE status should be returned otherwise.

4.2.3 When generated

This service primitive is generated when the actions begun by an IAPP-INITIATE.request are completed or the invocation of that primitive has failed.

4.2.4 Effect of receipt

Upon receipt of the IAPP-INITIATE.confirm(Status=SUCCESSFUL) corresponding to a previously issued IAPP-INITIATE.request, an APME should initialize the operation of the AP by issuing an 802.11 MLME-START.request primitive to the local 802.11 MLME. The APME should not issue an 802.11 MLME-START.request until an IAPP-INITIATE.confirm(Status=SUCCESSFUL) is received, i.e., to ensure that all associations in this BSS are reported to the ESS using IAPP, the AP should not begin operating until after the IAPP-INITIATE.confirm(Status=SUCCESSFUL) is received.

4.3 IAPP-TERMINATE.request

4.3.1 Function

This service primitive causes the IAPP entity to cease operation of the IAPP functions and protocol.

4.3.2 Semantics of the service primitive

The IAPP-TERMINATE.request primitive has the following semantics.

```
IAPP-TERMINATE.request {
                        }
```

4.3.3 When generated

This service primitive is generated by an APME when it is desired to terminate the operation of the IAPP entity. The APME should terminate operation of the local BSS, including disassociation of any associated STAs and ceasing of beacon transmissions, prior to terminating IAPP operation. The sole or final action by the APME in termination of local BSS operation should be issuance of an MLME-RESET.request. The IAPP-TERMINATE.request should be generated upon receipt of the corresponding MLME-RESET.confirm.

4.3.4 Effect of receipt

The UDP and TCP ports for the IAPP should be closed and the IAPP entity should cease operations.

4.4 IAPP-TERMINATE.confirm

4.4.1 Function

This service primitive notifies an APME that the actions begun by an IAPP-TERMINATE.request have been completed.

4.4.2 Semantics of the service primitive

The IAPP-TERMINATE.confirm primitive has the following semantics.

```
IAPP-TERMINATE.confirm {
                        Status
                        }
```

The Status parameter indicates the result of the corresponding IAPP-TERMINATE.request. The allowable value for the Status parameter is SUCCESSFUL.

4.4.3 When generated

This service primitive is generated by the IAPP entity when the actions begun by an IAPP-TERMINATE.request are completed.

4.4.4 Effect of receipt

Upon receipt of the IAPP-TERMINATE.confirm corresponding to a previously issued IAPP-TERMINATE.request, the APME should make no further service requests to the IAPP SAP without starting the IAPP entity again, using the IAPP-INITIATE.request primitive. Furthermore, the APME should not issue an MLME-START.request primitive prior to receipt of the subsequent IAPP-INITIATE.confirm primitive which indicates that the IAPP has been restarted successfully.

4.5 IAPP-ADD.request

4.5.1 Function

This service primitive is used when a STA associates with the AP using an 802.11 association request frame. The function of the IAPP-ADD.request primitive is two-fold. One purpose of this primitive is to cause the forwarding tables of layer 2 internetworking devices, e.g. bridges and switches, to be updated. This updates the layer 2 internetworking devices before a transmission from the associating STA, which might occur some arbitrary amount of time after the association. The second purpose of this primitive is to notify other APs within the multicast domain, i.e., that portion of a network in which a layer two frame addressed to a multicast address can be received, of the STA's new association, to allow those APs to clean up context information left behind by STAs that do not properly reassociate when moving from one AP to another, but rather only use the 802.11 Association Request.

4.5.2 Semantics of the service primitive

The IAPP-ADD.request primitive has the following semantics.

```
IAPP-ADD.request {
                    MAC Address,
                    Sequence Number,
                    Timeout
                    }
```

The MAC Address is the address of the STA that recently has successfully associated with the AP.

The Sequence Number is the value of the 802.11 Sequence Number field of the Association Request frame received from the associating STA. The sequence number is provided to aid the APME in other APs in the determination of whether the association represented by this IAPP-ADD.request is the most recent association for the STA identified by the MAC Address. The 802.11 sequence number may be an ambiguous indication of the most recent association. But, this information may be useful to an algorithm making a determination of the location of the most recent association of a STA.

The Timeout parameter is the value, in seconds that the IAPP-ADD.confirm primitive will be generated with a status of TIMEOUT, if both the ADD-notify packet (see 6.2) and the Layer 2 Update frame (see 6.3) have not been sent. The TIMEOUT status will not be generated by the IAPP-ADD.confirm only when both the ADD-notify packet and Layer 2 Update frame have been transmitted before the expiration of the period indicated by the Timeout parameter.

4.5.3 When generated

This service primitive should be generated by an APME when the local AP generates an 802.11 MLME-ASSOCIATE.indication.

4.5.4 Effect of receipt

Receipt of this service primitive should cause the following actions to occur:

1) The IAPP entity sends a Layer 2 Update frame to the DS, addressed such that it will cause forwarding tables in layer 2 devices that receive the frame to be updated so that all future traffic received by those layer 2 devices is forwarded to the port on which the frame was received,

2) The IAPP entity notifies the APs in the local multicast domain of the DS of the association between the AP and STA by sending an IAPP ADD-notify packet to the IAPP IP multicast address. The IAPP multicast address is 224.0.1.178. See RFC 1112.

3) The IAPP-ADD function also indicates that any cache entry for the STA must be cleared since a new association has occurred.

4.6 IAPP-ADD.confirm

4.6.1 Function

This service primitive is used to confirm that the actions initiated by an IAPP-ADD.request have been completed and inform an APME of the status of those actions.

4.6.2 Semantics of the service primitive

The IAPP-ADD.confirm primitive has the following semantics.

```
IAPP-ADD.confirm {
                    Status
                    }
```

The Status parameter indicates the success or failure of the corresponding IAPP-ADD.request. The allowable values for this parameter are SUCCESSFUL, FAIL and TIMEOUT. SUCCESSFUL status indicates that the corresponding IAPP-ADD.request was able to send both the IAPP ADD-notify packet and Layer 2 Update frame before the timeout expired. FAIL indicates that for some reason, the IAPP ADD-notify packet and the Layer2 Update frame could not be sent at all. TIMEOUT status indicates that one or both of the ADD-notify packet and Layer 2 Update frame were not sent before the timeout expired.

4.6.3 When generated

This service primitive is generated upon completion of the actions of the IAPP-ADD.request or expiration of the timeout specified in the corresponding IAPP-ADD.request primitive.

4.6.4 Effect of receipt

Upon receipt of this service primitive by an APME with Status=SUCCESSFUL, the APME should cause the DS Services to begin forwarding frames for the associated STA. Receipt of this primitive with Status=TIMEOUT should cause the APME to attempt to determine the cause of the failure to send the ADD-notify packet and Layer 2 Update frames and possibly invoke the IAPP-ADD.request again. When Status=FAIL, the STA's association should be denied or the STA disassociated.

4.7 IAPP-ADD.indication

4.7.1 Function

The IAPP-ADD.indication primitive is used to indicate to an APME that an association relationship has been established between a STA and another AP in the DS.

4.7.2 Semantics of the service primitive

The IAPP-ADD.indication primitive has the following semantics.

```
IAPP-ADD.indication {
                    MAC Address,
                    Sequence Number
                    }
```

The MAC Address is the address of the STA received in the IAPP ADD-notify packet.

The Sequence Number is the value of the 802.11 Sequence Number field of the Association Request frame received from the associating STA as received by the local IAPP entity in the ADD-notify packet. The sequence number is provided to aid the APME in the determination of whether the association represented by this IAPP-ADD.indication is the most recent association for the STA identified by the MAC Address. The 802.11 sequence number is not an unambiguous indication of the most recent association. But, this information may be useful to an algorithm making this determination.

4.7.3 When generated

This service primitive is generated upon receipt of an IAPP ADD-notify packet from the DS.

4.7.4 Effect of receipt

Upon receipt of this service primitive the APME should determine if the STA indicated by the MAC Address is shown to be associated with the AP receiving the IAPP-ADD.indication, with a sequence number older than that in the IAPP ADD-notify packet. If so, this service primitive should cause the generation of an 802.11 MLME-DISASSOCIATE.request by the APME. See 4.7.2. If the sequence number received in the IAPP ADD-notify packet is older than that received from the STA when it associated with the AP receiving the IAPP ADD-notify packet, the APME should ignore the indicated association and issue an IAPP-ADD.request.

Implementers of STA MAC entities are advised of the importance of continuing the sequential assignment of sequence numbers for outgoing MPDUs and MMPDUs throughout STA operation, as required by 802.11. A discontinuity in the sequence numbering at the time of reassociation could cause roaming in an IAPP environment to be unreliable. The 802.11 sequence number may be an ambiguous indication of the most recent association. But, this information may be useful to an algorithm making a determination of the location of the most recent association of a STA.

4.8 IAPP-MOVE.request

4.8.1 Function

This primitive should be issued by the APME when it receives an MLME-REASSOCIATE.indication from the MLME indicating that an STA has reassociated with the AP. It will attempt to send an IAPP MOVE-notify packet to the AP with which the reassociating STA was previously associated to notify that AP of the STA's reassociation.

4.8.2 Semantics of the service primitive

The IAPP-MOVE.request primitive has the following semantics.

```
IAPP-MOVE.request {
                    MAC Address,
                    Sequence Number,
                    Old AP,
                    Context Block,
                    Timeout
                    }
```

The MAC Address is the address of the STA that recently has successfully reassociated with the AP.

The Sequence Number is the value of the 802.11 Sequence Number field of the Reassociation Request frame received from the reassociating STA. The sequence number is provided to aid the APME in other APs in the determination of whether the association represented by this IAPP-MOVE.request is the most recent association for the STA identified by the MAC Address. The 802.11 sequence number is not an unambiguous indication of the most recent association. But, this information may be useful to an algorithm making this determination.

Old AP is the MAC address of the AP with which the reassociating STA was last associated. This value is obtained by the APME from the Current AP Address field of the 802.11 Reassociation Request frame.

The Context Block is the context to be sent to the Old AP. Otherwise, the Context Block is null. The Context Block is a container for information defined in 802.11 that is to be forwarded from one AP to another upon the reassociation of a STA.

The Timeout parameter value is the number of seconds expected for the IAPP MOVE-notify packet to be sent and the IAPP MOVE-response packet received. Failure to send the packet and receive a response in this time results in the IAPP-MOVE.confirm primitive being generated with a status of TIMEOUT.

4.8.3 When generated

This service primitive is generated by an APME when the MLME receives an 802.11 MLME-REASSOCIATE.indication from the local AP.

4.8.4 Effect of receipt

Receipt of this service primitive should cause the following actions to occur:

1) The IAPP entity determines the DSM layer 3 address of the AP identified by the old BSSID presented in the reassociation request and the security information needed to communicate with that AP using the methods described in clause 5.

2) The IAPP entity requests any context stored at the AP with which the STA was previously associated to be forwarded to the AP with which the STA is currently associated by sending an IAPP MOVE-notify packet to the old AP.

4.8.5 Utilization of Proactive Caching

If the APME is utilizing caching, then the APME should first lookup the STA's context in the IAPP cache using the STA's MAC Address. If found (a cache hit), then an IAPP-MOVE.request does not have to be issued until after an 802.11 Reassociation Response frame. If the STA context is not found in the cache (a cache miss), then the APME should issue an IAPP-MOVE.request as usual. Furthermore, the MAC Address of the Old AP (obtained from the 802.11 Reasociation Request frame) is added to the neighbor graph of the APME.

4.9 IAPP-MOVE.confirm

4.9.1 Function

This service primitive is used to confirm that the actions initiated by an IAPP-MOVE.request have been completed and inform an APME of the status of those actions.

4.9.2 Semantics of the service primitive

The IAPP-MOVE.confirm primitive has the following semantics.

```
IAPP-MOVE.confirm {
                    MAC Address,
```

Sequence Number,
Old AP,
New BSSID,
Context Block,
Status
}

The MAC Address is the address of the STA from the corresponding IAPP-MOVE.request.

The Sequence Number is the value of the 802.11 Sequence Number field of the Reassociation Request frame received from the reassociating STA.

Old AP is the MAC address of the AP with which the reassociating STA was last associated. This value is obtained by the IAPP from the received MOVE-notify packet

The New BSSID parameter is the WM MAC address of the AP with which the STA has reassociated.

The Context Block is the context returned by the Old AP, if the Status is SUCCESSFUL. Otherwise, the Context Block is null. The Context Block is a container for information defined by other 802.11 standards that is to be forwarded from one AP to another upon the reassociation of a STA. If the Old AP does not return any context information, the Context Block can be null, even when the status is SUCCESSFUL.

The Status parameter indicates the result of the corresponding IAPP-MOVE.request. The allowable values for this parameter are SUCCESSFUL, STALE_MOVE, MOVE_DENIED, FAIL, and TIMEOUT. The TIMEOUT status indicates the corresponding IAPP-MOVE.request primitive was not able to complete the transmission of both the IAPP MOVE-notify packet and IAPP Layer 2 Update frame, as well as receive the IAPP MOVE-response packet before the timeout parameter of the IAPP-MOVE.request primitive expired. The STALE_MOVE status indicates that the corresponding IAPP-MOVE.request did not complete successfully, because the IAPP MOVE-response packet returned by the Old AP contained a status value indicating a stale move. MOVE_DENIED indicates that the AP receiving the IAPP-MOVE.indication either is not able to verify a previous association by the indicated STA or has some other reason to deny the reassociation at the AP that sent the IAPP Move-notify packet. FAIL indicates that a RADIUS Access-Reject was received in response to the RADIUS Access-Request sent to the RADUS server to look up the IP address of the Old AP.

4.9.3 When generated

This service primitive is generated upon receipt of context information from the Old AP in an IAPP MOVE-response packet as a result of the Old AP's use of the IAPP-MOVE.response primitive or expiration of the timeout specified in the corresponding IAPP-MOVE.request primitive. If the Status is SUCCESSFUL, the IAPP entity sends a Layer 2 Update frame to the DS, addressed such that it will cause forwarding tables in layer 2 devices that receive the frame to be updated so that all future traffic received by those layer 2 devices is forwarded to the port on which the frame was received,

4.9.4 Effect of receipt

Upon receipt of this service primitive by an APME with SUCCESSFUL status, the APME should cause the DS services to begin forwarding frames for the reassociated STA. Completion of the IAPP-MOVE.request includes receipt of STA context from the Old AP, when the Status is SUCCESSFUL. When the Status is not SUCCESSFUL, the APME should disassociate the STA indicated by the MAC Address parameter, using the 802.11 MLME-DISASSOCIATE.request primitive with a Reason Code of 1, meaning "Unspecified Reason". Future revisions of the IEEE Std 802.11 may define a new Reason Code that means "Old AP did not verify previous association." Should this Reason Code be defined, it should be used in preference to Reason Code 1.

4.10 IAPP-MOVE.indication

4.10.1 Function

This service primitive is used to indicate that a STA has reassociated with another AP.

4.10.2 Semantics of the service primitive

The IAPP-MOVE.indication primitive has the following semantics.

```
IAPP-MOVE.indication {
                    MAC Address,
                    New BSSID
                    Sequence Number,
                    AP Address,
                    Context Block
                    }
```

The MAC Address is the address of the STA that has reassociated with the AP that sent the IAPP MOVE-notify packet.

The New BSSID parameter is the WM MAC address of the AP sending the IAPP MOVE-notify packet.

The Sequence Number is the value of the 802.11 Sequence Number field of the Reassociation Request frame received from the reassociating STA. The sequence number is provided to aid the APME in the determination of whether the association represented by this IAPP-MOVE.indication is the most recent association for the STA identified by the MAC Address. The 802.11 sequence number is not an unambiguous indication of the most recent association. But, this information may be useful to an algorithm making this determination.

The AP Address is the DSM IP address of the AP sending the IAPP MOVE-notify packet.

The Context Block is the context sent by the AP indicated by the AP Address. Otherwise, the Context Block is null. The Context Block is a container for information defined by other 802.11 standards that is to be forwarded from one AP to another upon the reassociation of a STA.

4.10.3 When generated

This service primitive is generated when an IAPP MOVE-notify packet is received.

4.10.4 Effect of receipt

Upon receipt of this service primitive with a sequence number indicating a more recent association than that at the receiving AP (if any), the AP should forward any relevant context related to the reassociated STA to the AP with which the STA is now associated by using the IAPP-MOVE.response primitive and process any context received in the Context Block received and should issue an MLME-DISASSOCIATION.request for the STA indicated by the MAC Address parameter. "Relevant" context for a STA is defined as those information elements that other 802.11 standards require to be forwarded when a STA reassociates. If the received sequence number does not represent a more recent association than that at the AP where the IAPP-MOVE.indication is received, the APME should ignore the indicated reassociation, the APME should issue an IAPP-MOVE.response with a status of STALE_MOVE that will cause an IAPP MOVE-response packet to be sent to the AP that originated the IAPP MOVE-notify packet, and the APME should issue an IAPP-ADD.request primitive of its own to ensure that all layer 2 devices are properly informed of the correct location of the STA's most recent association.

4.11 IAPP-MOVE.response

4.11.1 Function

This service primitive is used to send any relevant context resident in the AP issuing this primitive to another AP when a STA has reassociated with that other AP. "Relevant" context for a STA is defined as those information elements that other 802.11 standards require to be forwarded when a STA reassociates.

4.11.2 Semantics of the service primitive

The IAPP-MOVE.response primitive has the following semantics.

```
IAPP-MOVE.response {
                    MAC Address,
                    Sequence Number,
                    AP Address,
                    Context Block,
                    Status
                    }
```

The MAC Address is the address of the STA that has reassociated with the AP identified by the AP Address.

The Sequence Number is the value of the 802.11 Sequence Number field of the Reassociation Request frame received from the reassociating STA.

The AP Address is the DSM IP address of the AP where the STA has reassociated.

The Context Block is the context for the reassociated STA. The Context Block may be null.

The Status parameter indicates the result of the corresponding IAPP-MOVE.indication. The allowable values for this parameter are SUCCESSFUL, MOVE_DENIED, and STALE_MOVE. STALE_MOVE should be used to indicate that the AP receiving the IAPP-MOVE.indication has a current association with the STA indicated by the MAC Address parameter with a more recent sequence number than that in the IAPP-MOVE.indication. MOVE_DENIED should be used to indicate that the AP receiving the IAPP-MOVE.indication either is not able to verify a previous association by the indicated STA or has some other reason to deny the reassociation at the AP that sent the IAPP Move-notify packet.

4.11.3 When generated

This service primitive should be generated by the APME when an IAPP-MOVE.indication is received.

4.11.4 Effect of receipt

Upon receipt of this service primitive, the AP forwards all relevant context related to the reassociated STA and the Status to the peer IAPP entity in the AP with which the STA is now associated by sending the IAPP MOVE-response packet. Any context for the STA identified by the MAC Address parameter may be discarded upon issuance of this response.

4.12 IAPP-CACHE-NOTIFY.request

4.12.1 Function

This primitive is used by the APME when caching is enabled and the APME receives an MLME-REASSOCIATE.indication or a MLME-ASSOCIATE.indication from the MLME indicating that the STA has reassociated or associated with the AP. This primitive causes the IAPP entity to send IAPP CACHE-notify packets to each of the APs in the neighbor graph requesting the included context to be cached. Receipt of the IAPP CACHE-notify packets at the neighboring APs causes the IAPP to issue an IAPP-CACHE-NOTIFY.indication primitive at the neighboring APs.

4.12.2 Semantics of the service primitive

The IAPP-CACHE-NOTIFY.request primitive has the following semantics.

```
IAPP-CACHE-NOTIFY.request {
                    MAC Address,
                    Sequence Number,
                    Current AP,
                    Context Block,
                    ContextTimeout,
                    RequestTimeout,
                    }
```

The MAC Address is the address of the STA that recently has successfully associated or reassociated with the AP.

The Sequence Number is the value of the 802.11 Sequence Number field of the Association Request or Reassociation Request frame received from the STA. The sequence number is provided to aid the APME in other APs in the determination of whether the association represented by this IAPP-CACHE-NOTIFY.request is the most recent association for the STA identified by the MAC Address. The 802.11 sequence number is not an unambiguous indication of the most recent association. But, this information may be useful to an algorithm making this determination.

Current AP is the MAC address of the AP with which the STA is currently associated.

The Context Block is the context to be sent to the neighbor AP. Otherwise, the Context Block is null. The Context Block is a container for information defined in 802.11 that is to be forwarded from one AP to another in anticipation of the reassociation of a STA.

The ContextTimeout parameter value is the number of seconds for which the neighboring AP should maintain the STA Context before removing it from the cache. Receipt of an IAPP-CACHE-NOTIFY.indication should refresh the cache and reset the ContextTimeout value.

The RequestTimeout parameter value is the number of seconds that the IAPP entity has for the IAPP CACHE-notify packets to be sent and the IAPP CACHE-response packets received. Failure to send the packets and receive a responses in this time results in the IAPP-CACHE-NOTIFY.confirm primitive being issued to the APME with a status of TIMEOUT.

4.12.3 When Generated

This service primitive is generated by the APME upon receipt of an MLME-REASSOCIATE.indication or a MLME-ASSOCIATE.indication from the local AP, or when the APME detects that STA context has changed. Detection of a STA context change is context dependent.

4.12.4 Effect of receipt

Receipt of this service primitive should cause the IAPP entity to send IAPP CACHE-notify packets to each of the APs in its neighbor graph. The IAPP entity is responsible for reliable delivery to each of the APs in the neighboring AP graph. In addition, the IAPP entity should send a Layer 2 Update frame with the MAC address of the STA as the source address.

4.13 IAPP-CACHE-NOTIFY.confirm

4.13.1 Function

This service primitive is used to confirm that the actions initiated by an IAPP-CACHE-NOTIFY.request has been completed and inform an APME of the status of that action. It is also issued upon expiration of the timeout of the IAPP-CACHE-NOTIFY.request {RequestTimeout} value.

4.13.2 Semantics of the service primitive

The IAPP-CACHE-NOTIFY.confirm primitive has the following semantics.

```
IAPP-CACHE-NOTIFY.confirm {
                MAC Address,
                Sequence Number,
                Status
                }
```

The MAC Address is the value of the MAC Address from the corresponding IAPP-CACHE-NOTIFY.request.

The Sequence Number is the value of the Sequence Number from the corresponding IAPP-CACHE-NOTIFY.request.

The Status parameter indicates the result of the corresponding IAPP-CACHE-NOTIFY.request. The allowable values for this parameter are SUCCESSFUL, STALE_CACHE, and TIMEOUT. The TIMEOUT status indicates the IAPP was not able to complete the transmission of the IAPP CACHE-notify packets to any of its neighboring APs, as well as receive the IAPP CACHE-response packets from those neighboring APs before the RequestTimeout parameter of the IAPP-CACHE-NOTIFY.request primitive expired. STALE_CACHE indicates that at least one neighboring AP responded with an IAPP CACHE-response packet containing a Status field with a value of STALE_CACHE. SUCCESSFUL indicates that at least one neighboring AP responded with an IAPP CACHE-response packet containing a Status field with a value of SUCCESSFUL and no neighboring APs responded with an IAPP CACHE-response packet containing a Status field with a value of STALE_CACHE.

4.13.3 When generated

This service primitive is generated upon receipt of all the CACHE-response packets from each of the neighboring APs or a timeout.

4.13.4 Effect of receipt

Upon receipt of this service primitive by an APME with SUCCESSFUL status indicates that all the neighbor APs responded with status of SUCCESSFUL in the CACHE-response packet. This indicates that the station context cache is fresh and the neighbor graph is current. Receipt of this service primitive with STALE_CACHE status indicates that at least one neighboring AP responded with status of STALE_CACHE in its CACHE-response. In this case, the APME should delete its station cache entry for the station indicated by the MAC Address parameter. Receipt of this service primitive with TIMEOUT status indicates that the IAPP did not receive any responses from neighboring APs before the expiration of the RequestTimeout parameter of the corresponding IAPP-CACHE-NOTIFY.request primitive.

4.14 IAPP-CACHE-NOTIFY.indication

4.14.1 Function

This service primitive is used to indicate that a CACHE-notify packet has been received by this AP from one of its neighbors. The IAPP CACHE-NOTIFY.indication contains STA context information that should be updated or added to this AP's neighboring STA context cache.

4.14.2 Semantics of the service primitive

The IAPP-CACHE-NOTIFY.indication primitive has the following semantics.

```
IAPP-CACHE-NOTIFY.indication {
                MAC Address,
                Sequence Number,
                Current AP,
```

Context Block,
Context Timeout
}

The MAC Address is the value of the MAC Address field from the CACHE-notify packet.

The Sequence Number is the value of the Sequence Number field from the CACHE-notify packet. The sequence number is provided to aid the APME in the determination of whether the association indicated by this primitive is the most recent association for the STA identified by the MAC Address. The 802.11 sequence number is not an unambiguous indication of the most recent association. But, this information may be useful to an algorithm making this determination.

The Cureent AP is the value of the Current AP field of the IAPP CACHE-notify packet.

The Context Block is the context received in the CACHE-notify packet. Otherwise, the Context Block is null. The Context Block is a container for information defined in 802.11 that is to be forwarded from one AP to another in anticipation of the reassociation of a STA.

The Context Timeout is the value of the Context Timeout field from the CACHE-notify packet. Receipt of an IAPP-CACHE-UPDATE.request should refresh the cache and reset the ContextTimeout value.

4.14.3 When Generated

This service primitive is generated by the IAPP entity upon receipt of a CACHE-notify packet from a neighboring AP.

4.14.4 Effect of receipt

Receipt of this service primitive should cause the APME to cache any context as described in sections 5.6.1 and 5.6.2. When this action is completed, the APME should send an IAPP-CACHE-NOTIFY.response primitive to the IAPP entity.

4.15 IAPP-CACHE-NOTIFY.response

4.15.1 Function

This service primitive is used to indicate that a CACHE-response packet has been received.

4.15.2 Semantics of the service primitive

The IAPP-CACHE-NOTIFY.response primitive has the following semantics.

IAPP-CACHE-NOTIFY.response {
MAC Address,
Sequence Number,
Status
}

The MAC Address is the value of the MAC Address field from the corresponding IAPP-CACHE-NOTIFY.indication primitive.
The Sequence Number is the value of the Sequence Number field from the corresponding IAPP-CACHE-NOTIFY.indication primitive.

The Status parameter may have either of two values, SUCCESSFUL or STALE_CACHE. The value SUCCESSFUL indicates that the station context cache was updated as requested by the corresponding IAPP-CACHE-NOTIFY.indication primitive. The value STALE_CACHE indicates that the station context cache was not updated as requested, because the local value in the cache is more recent than that indicated by the corresponding IAPP-CACHE-NOTIFY.indication primitive.

4.15.3 When Generated

This service primitive is invoked by the APME when the cache addition or update as requested by the IAPP-CACHE-NOTIFY.indication primitive has been completed.

4.15.4 Effect of receipt

Receipt of this service primitive causes the IAPP entity to send a CACHE-response packet to the neighboring AP that sent the CACHE-notify packet that caused the corresponding IAPP-CACHE-NOTIFY.indication to be issued.

4.16 Message Sequence Charts

Figure 3 through Figure 18 in this section are message sequence charts showing the relationship of the service primitives and information exchanges between layer entities and peer layers.

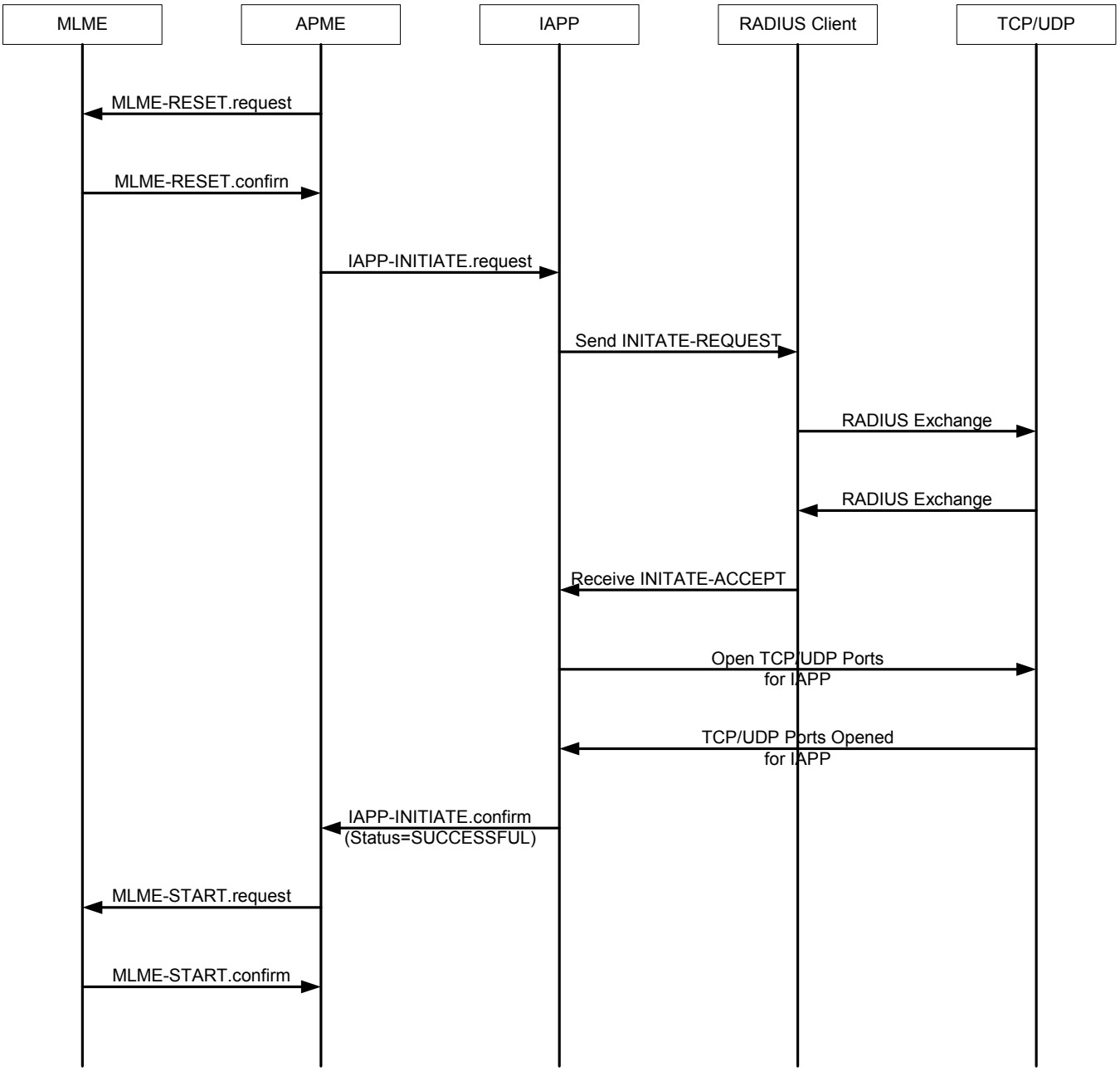

Figure 3, Normal initiation of the IAPP protocol

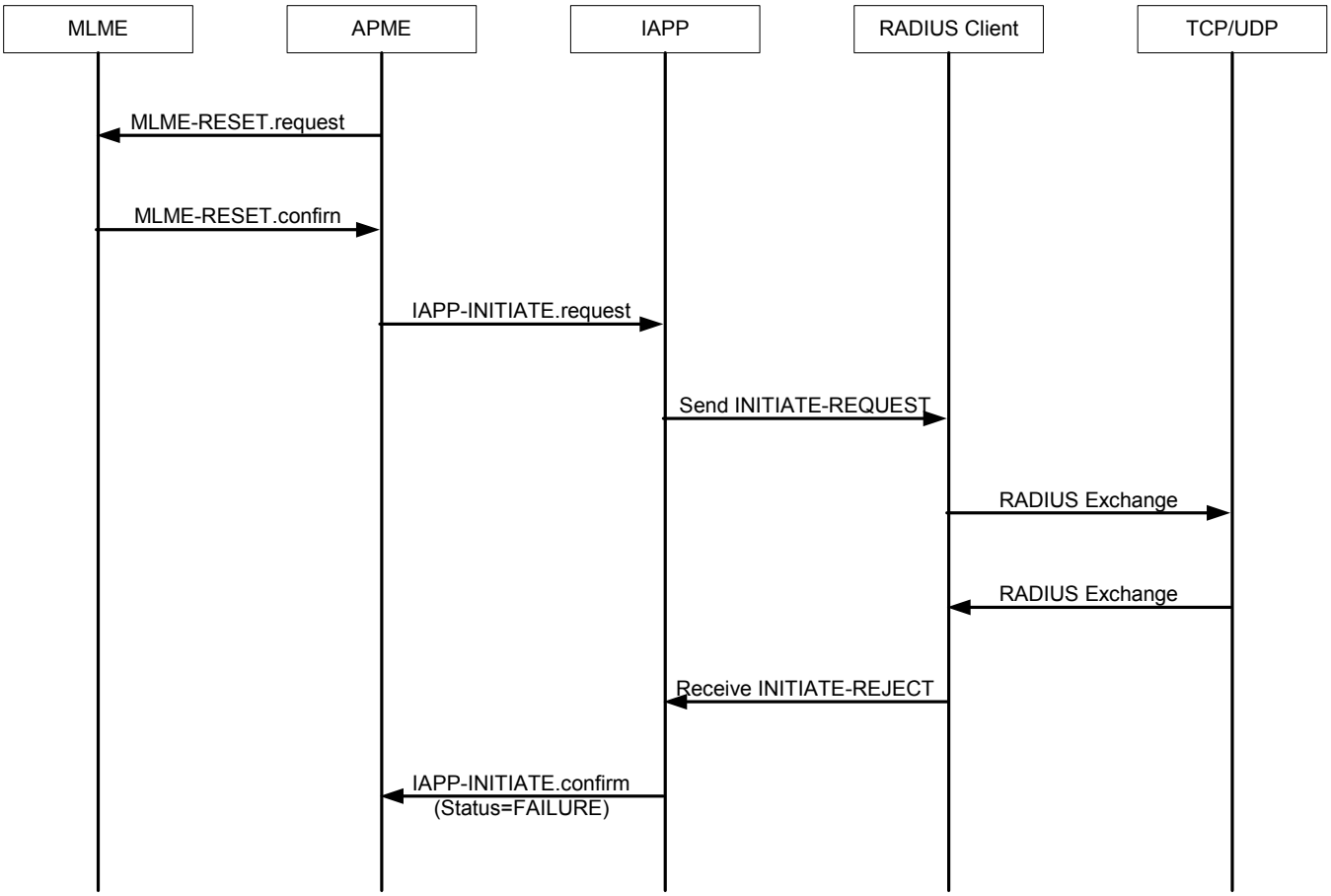

Figure 4, Failed initiation of the IAPP Protocol

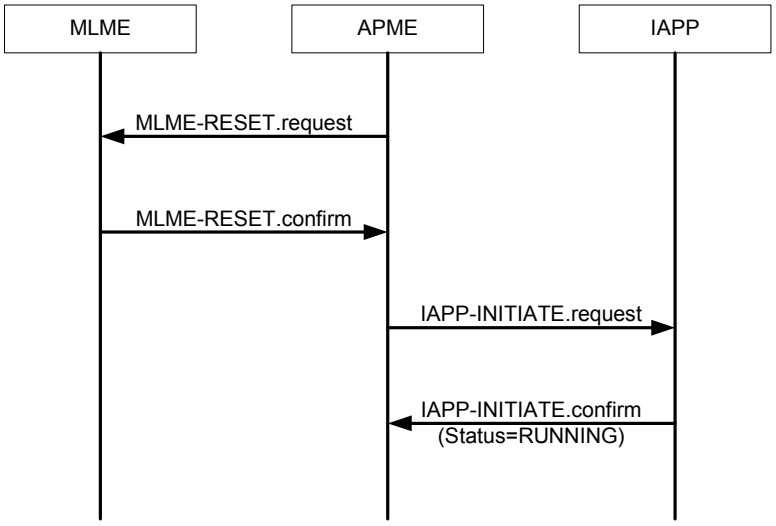

Figure 5, Attempted re-initiation of the IAPP protocol

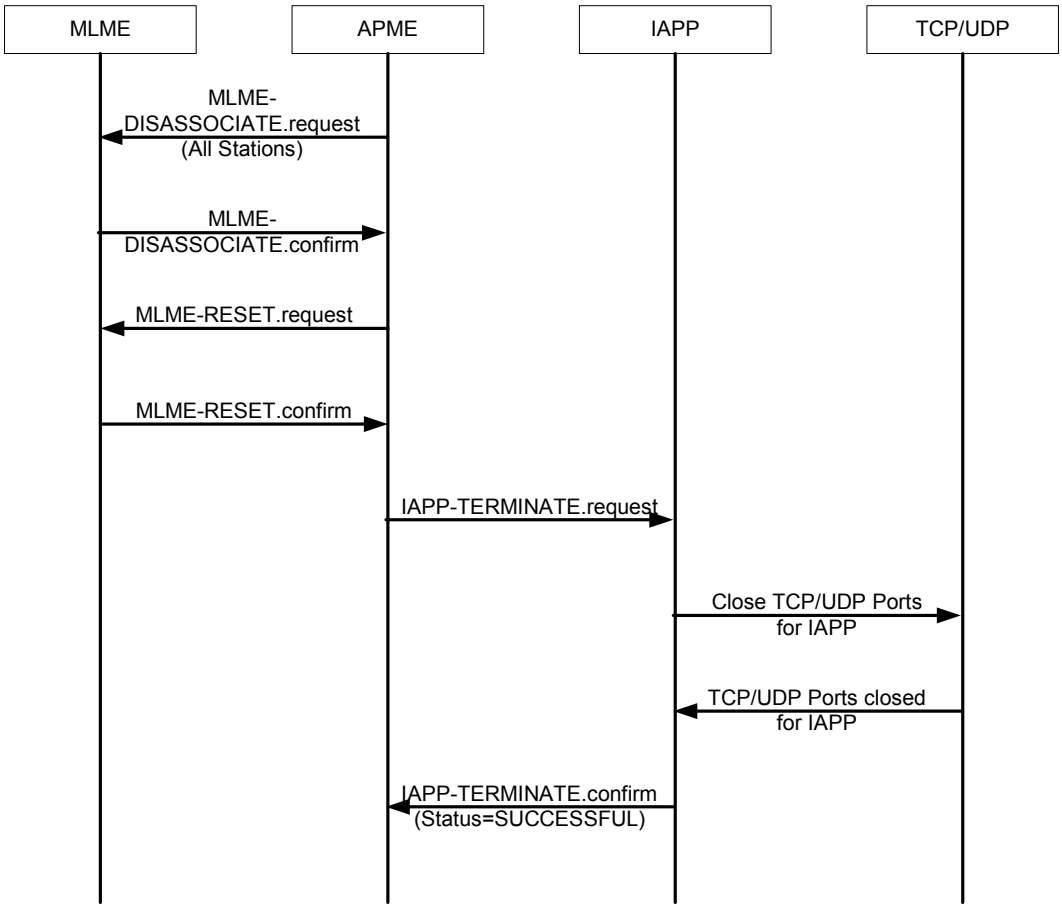

Figure 6, Termination of the IAPP protocol

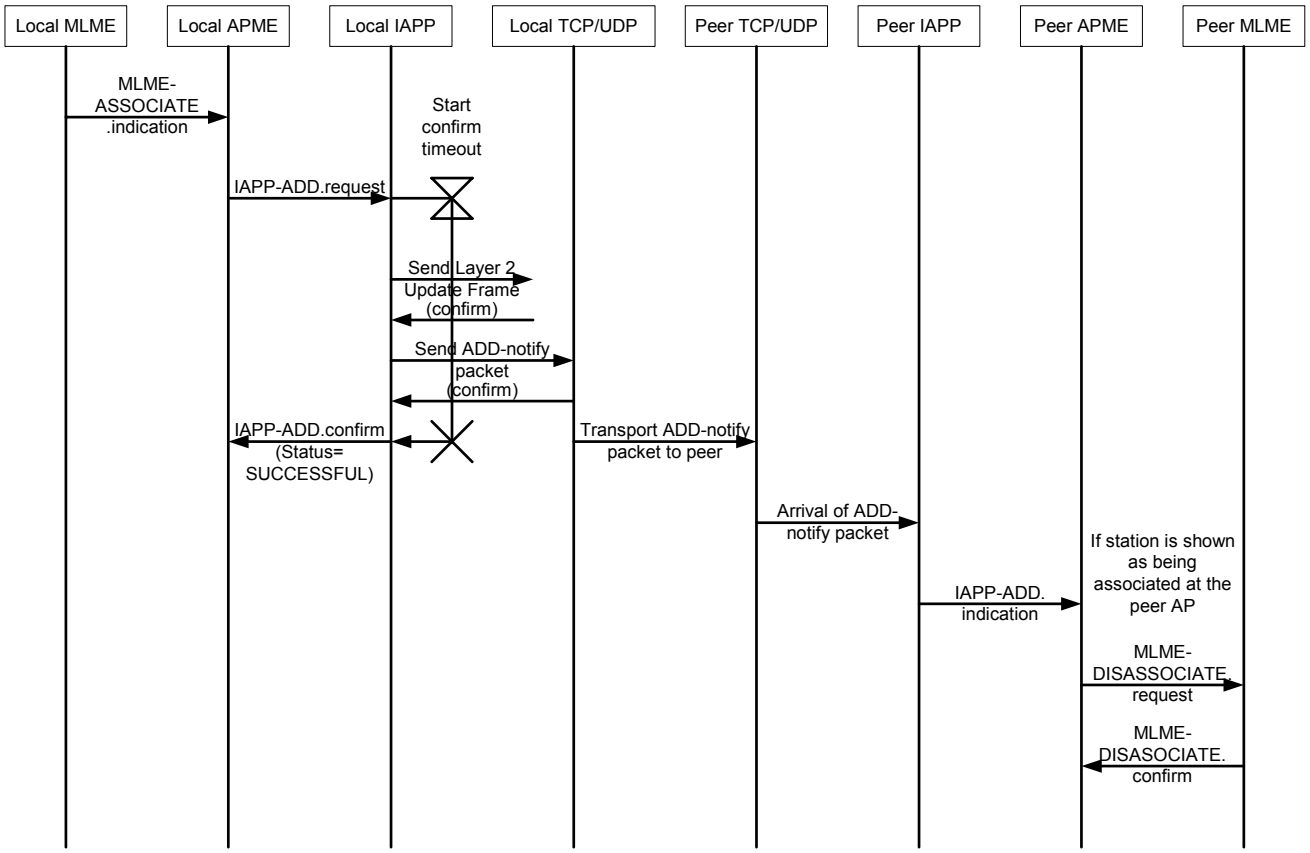

Figure 7, STA association

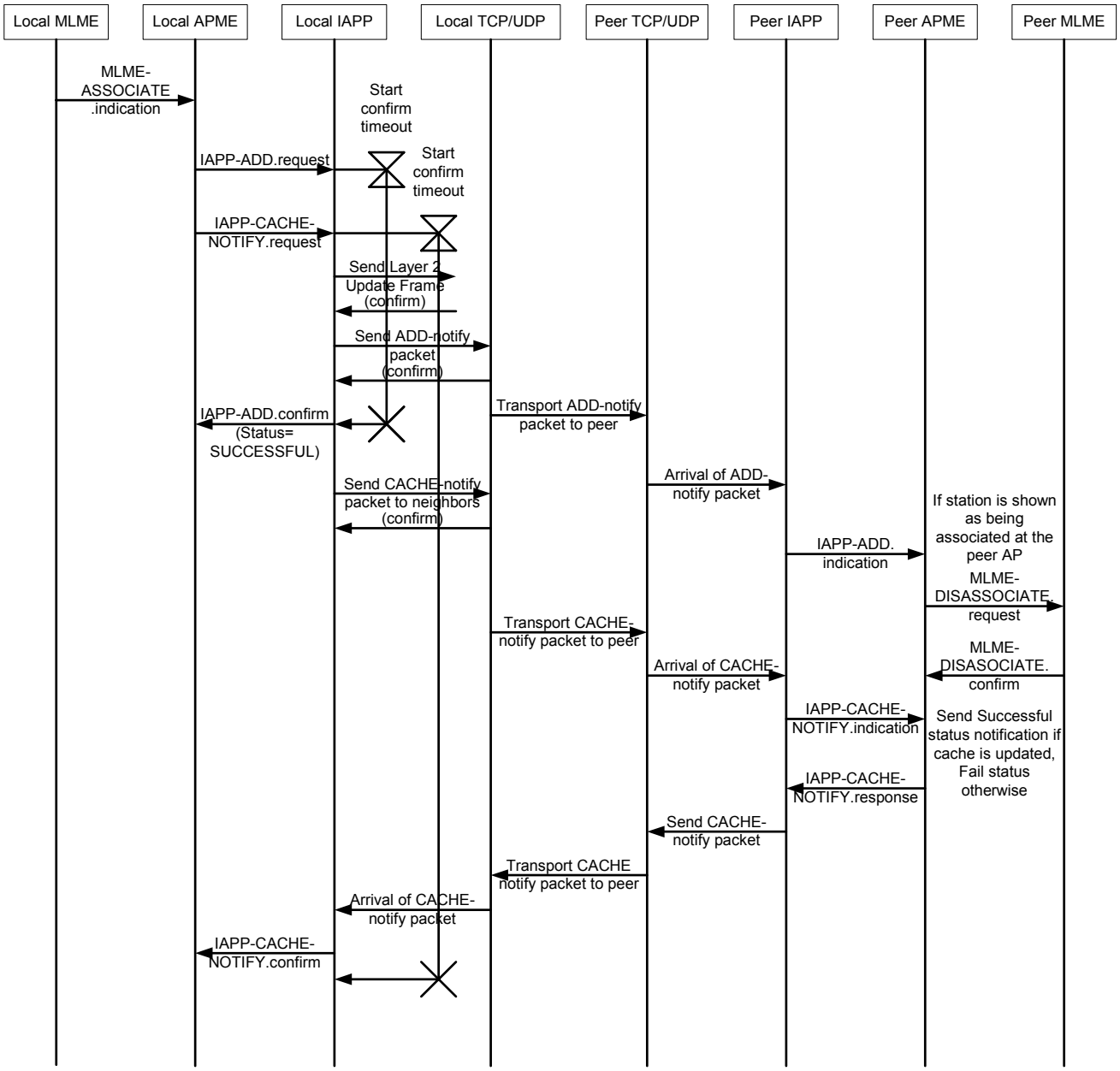

Figure 8, STA association – caching enabled

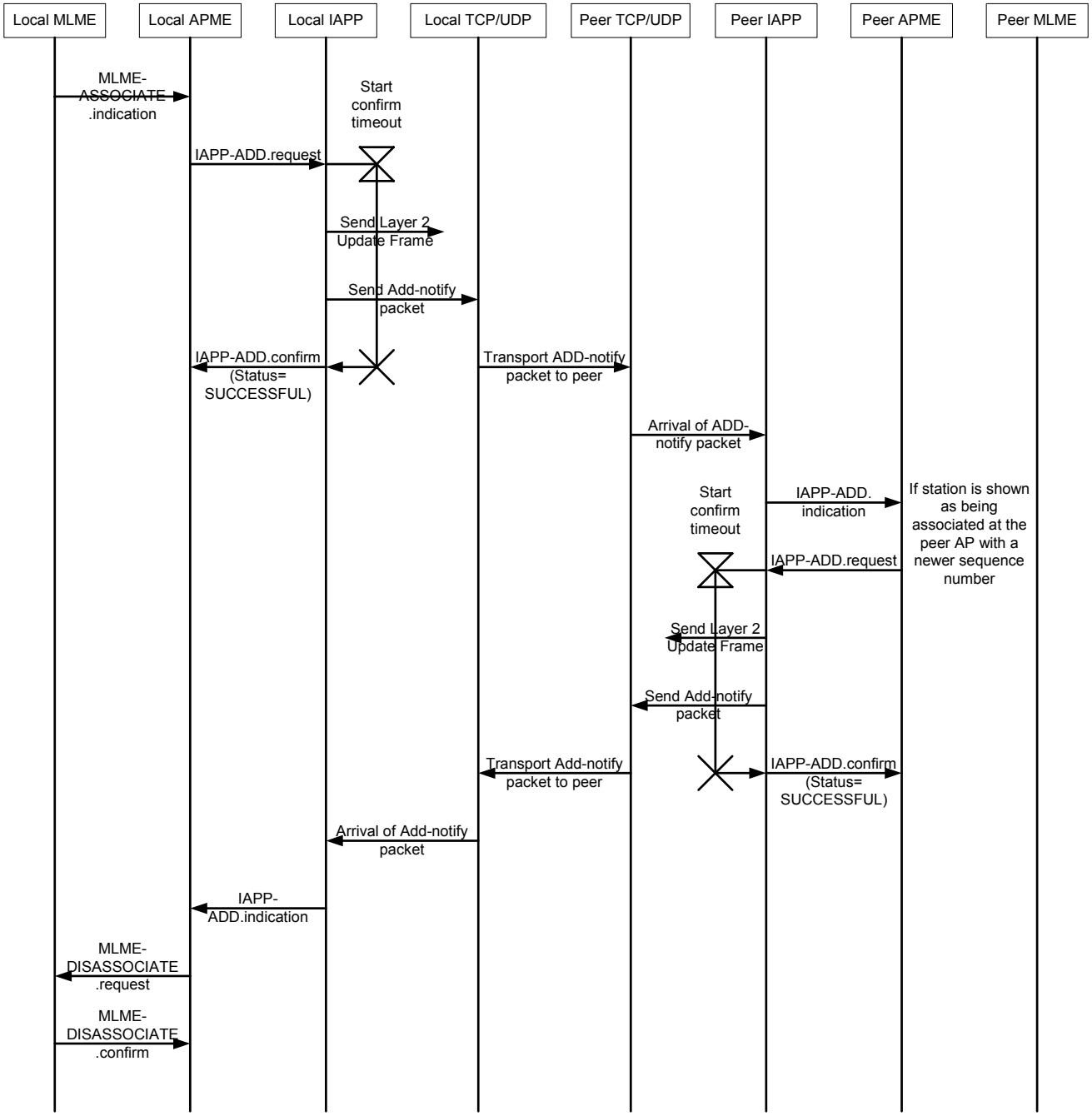

Figure 9, STA association - stale association

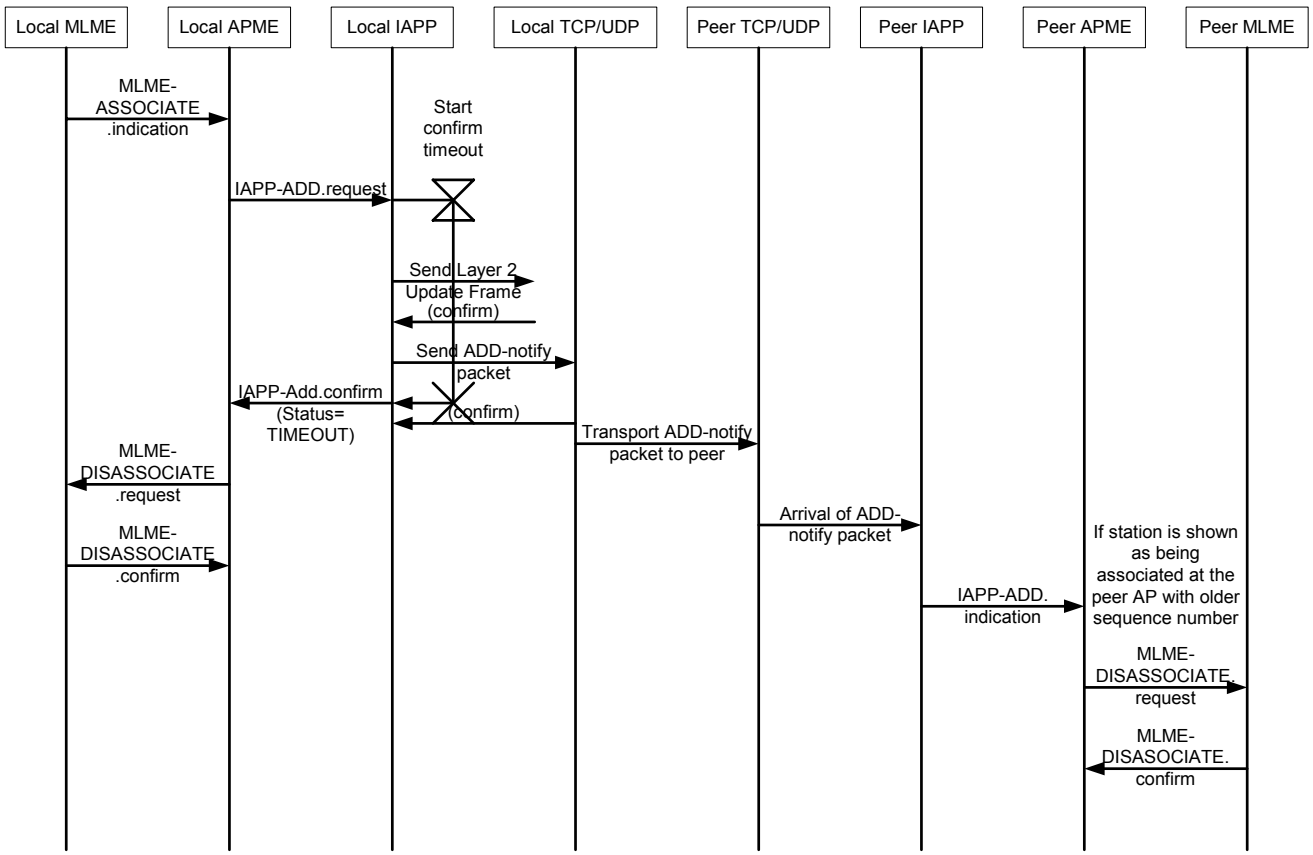

Figure 10, STA association – timeout

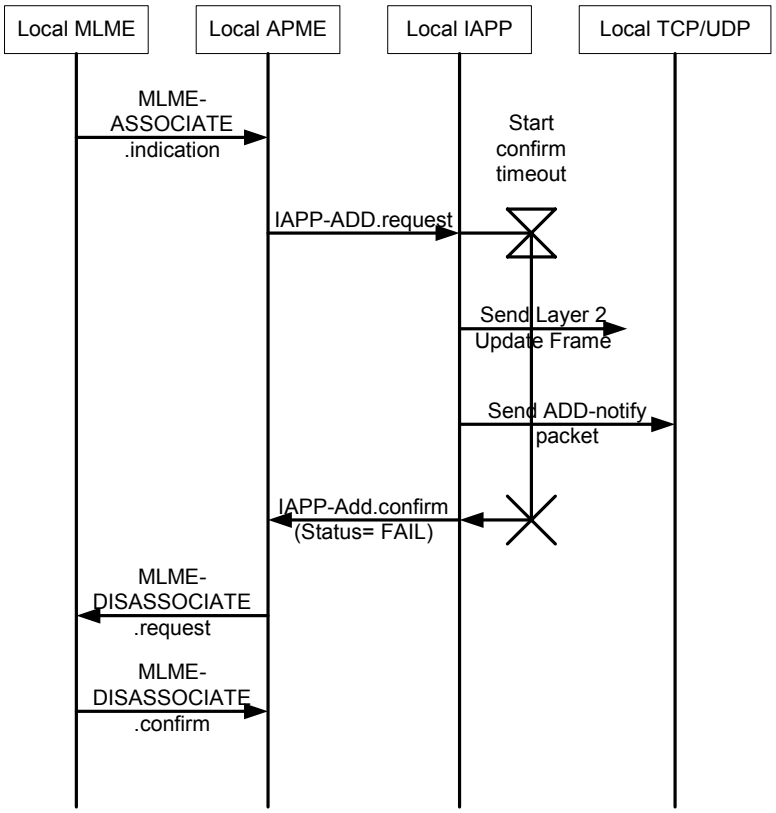

Figure 11, STA association – Failure

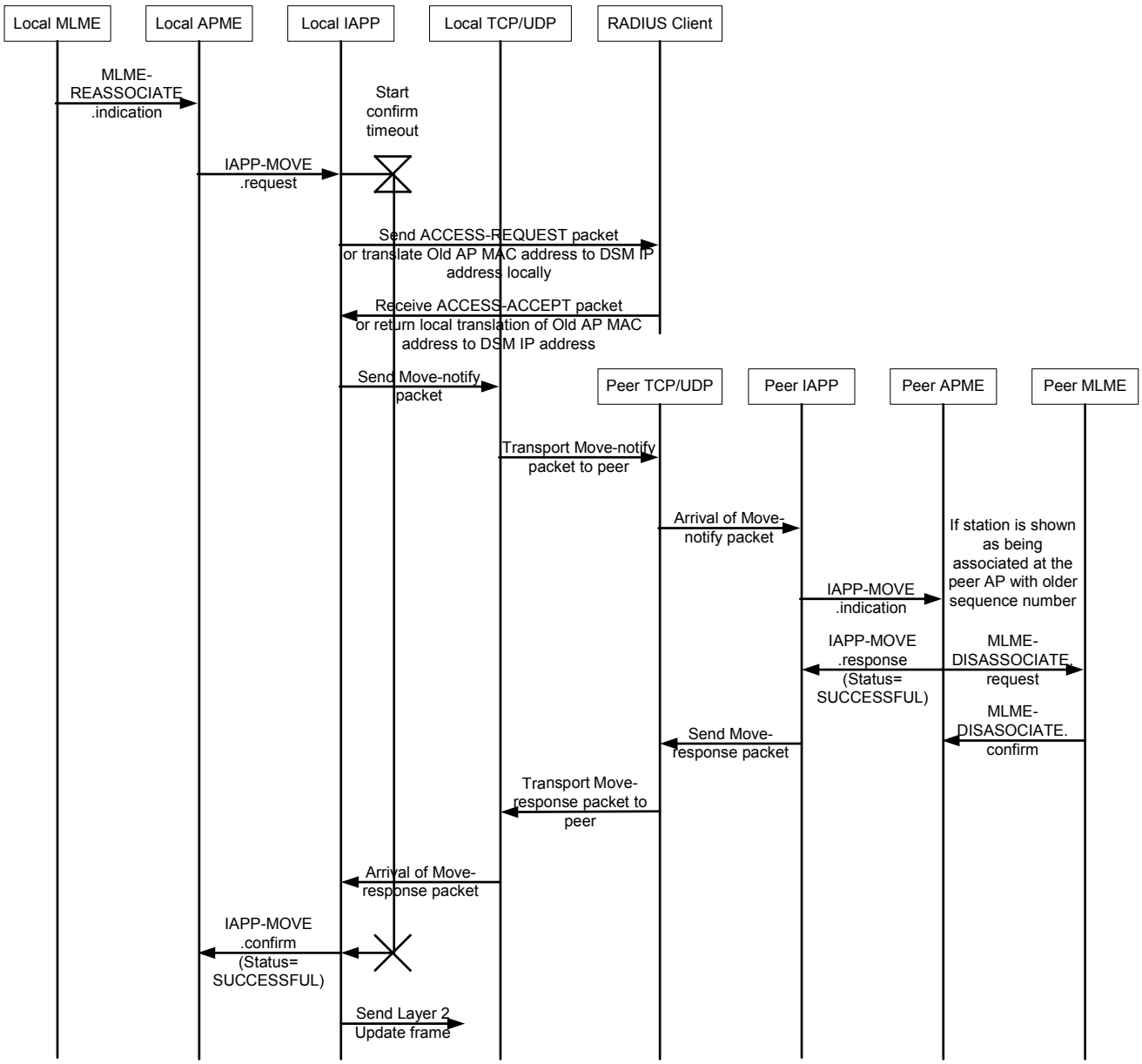

Figure 12, STA reassociation

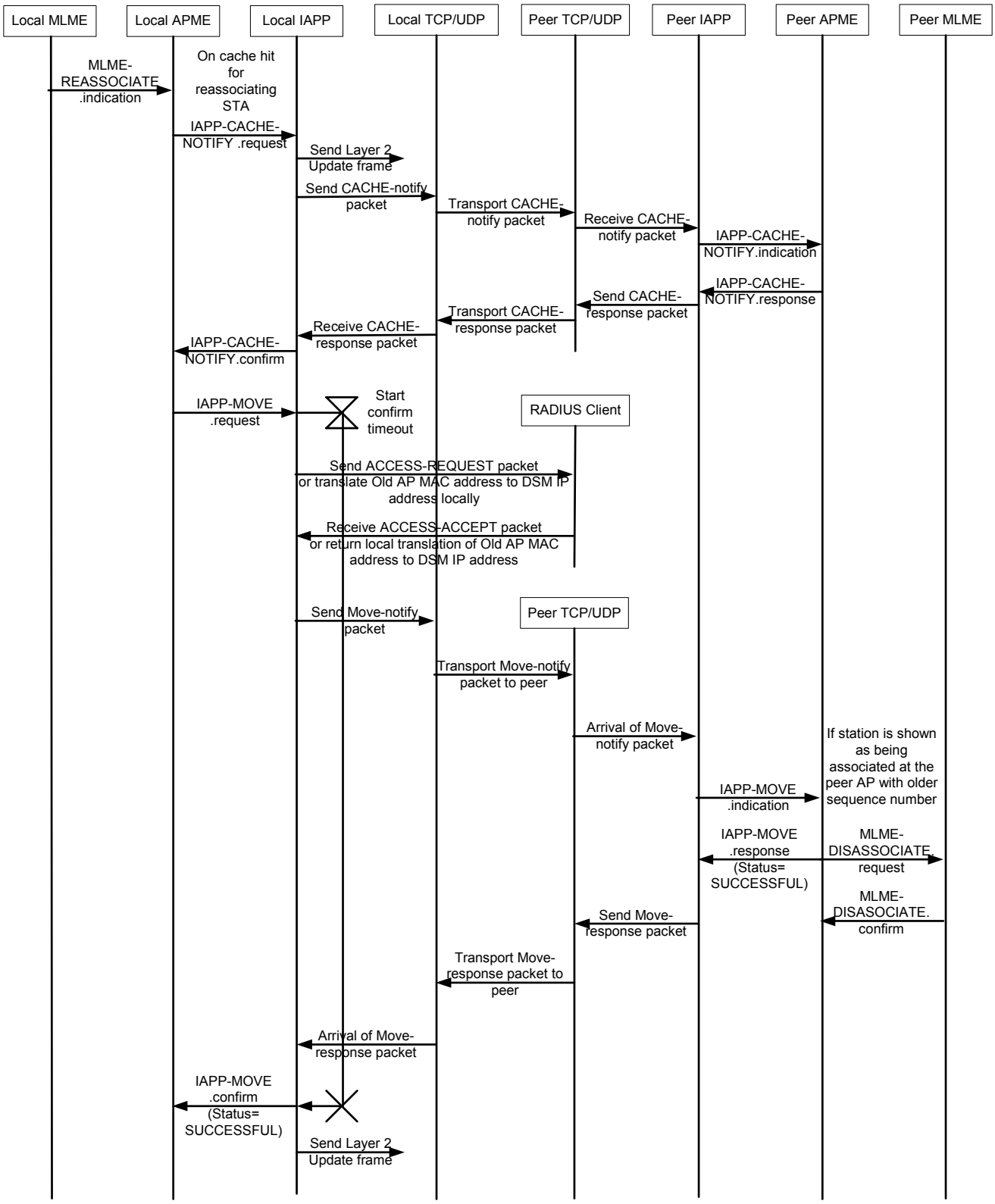

Figure 13 – STA reassociation using caching (cache hit)

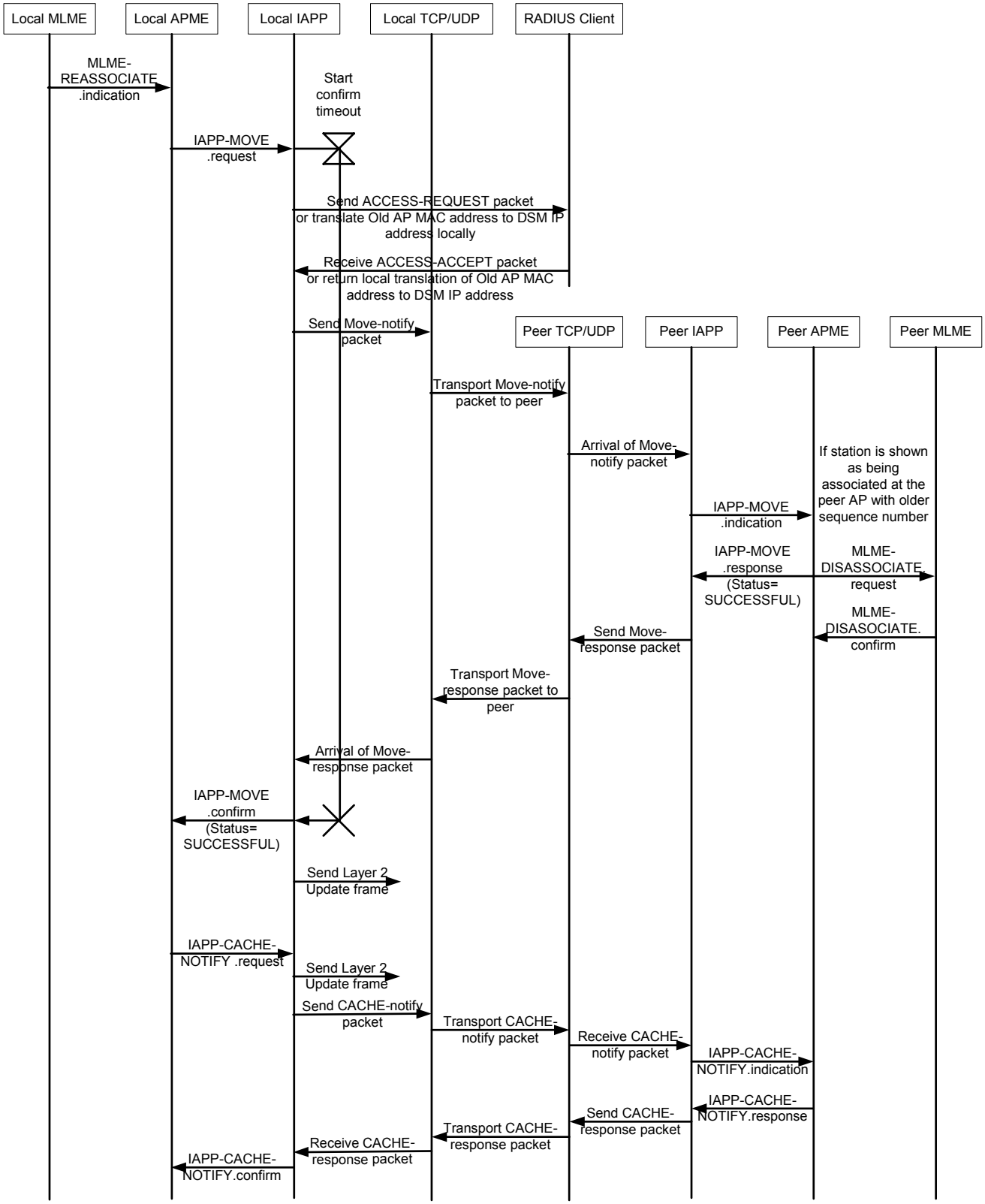

Figure 14, STA reassociation with caching enabled (cache miss)

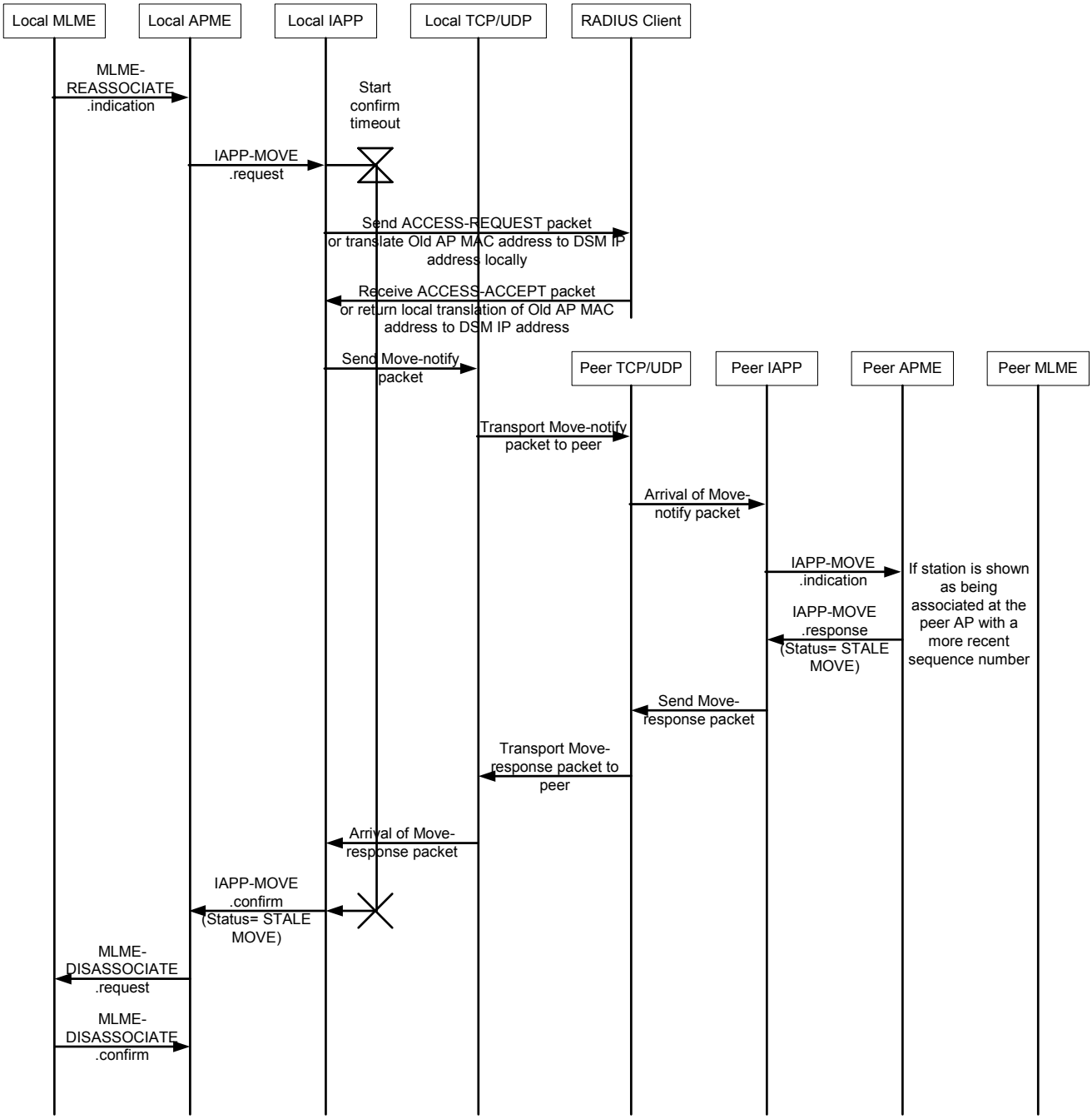

Figure 15, STA reassociation - stale move

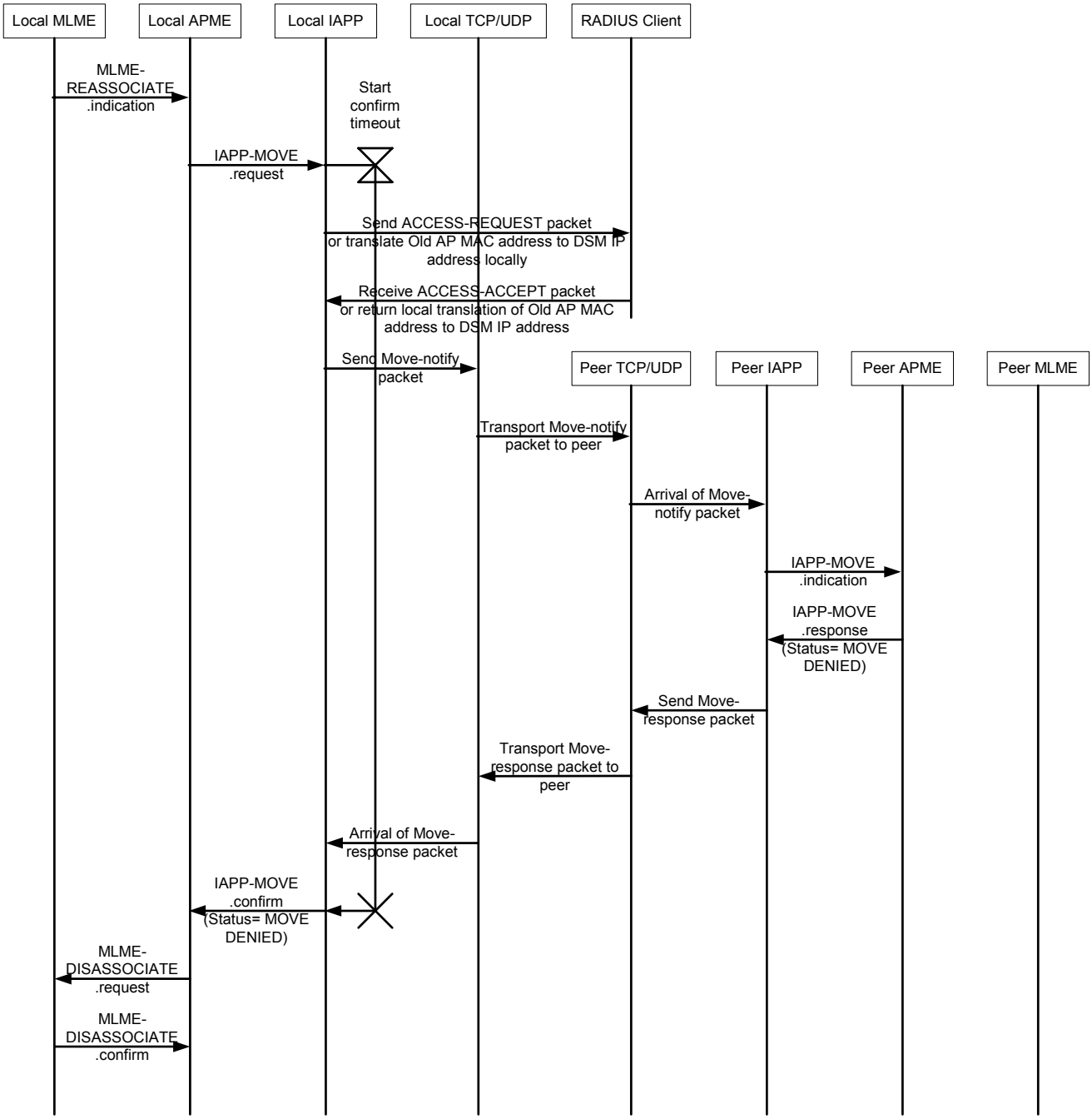

Figure 16, STA reassociation - move denied

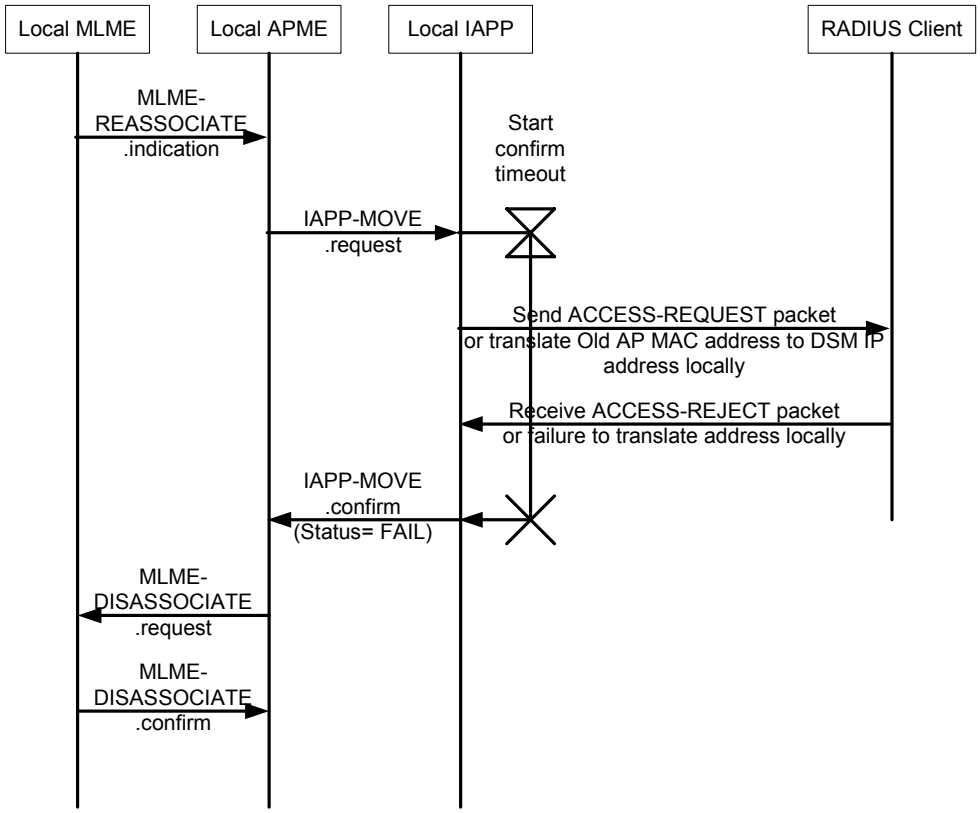

Figure 17, STA reassociation – failure

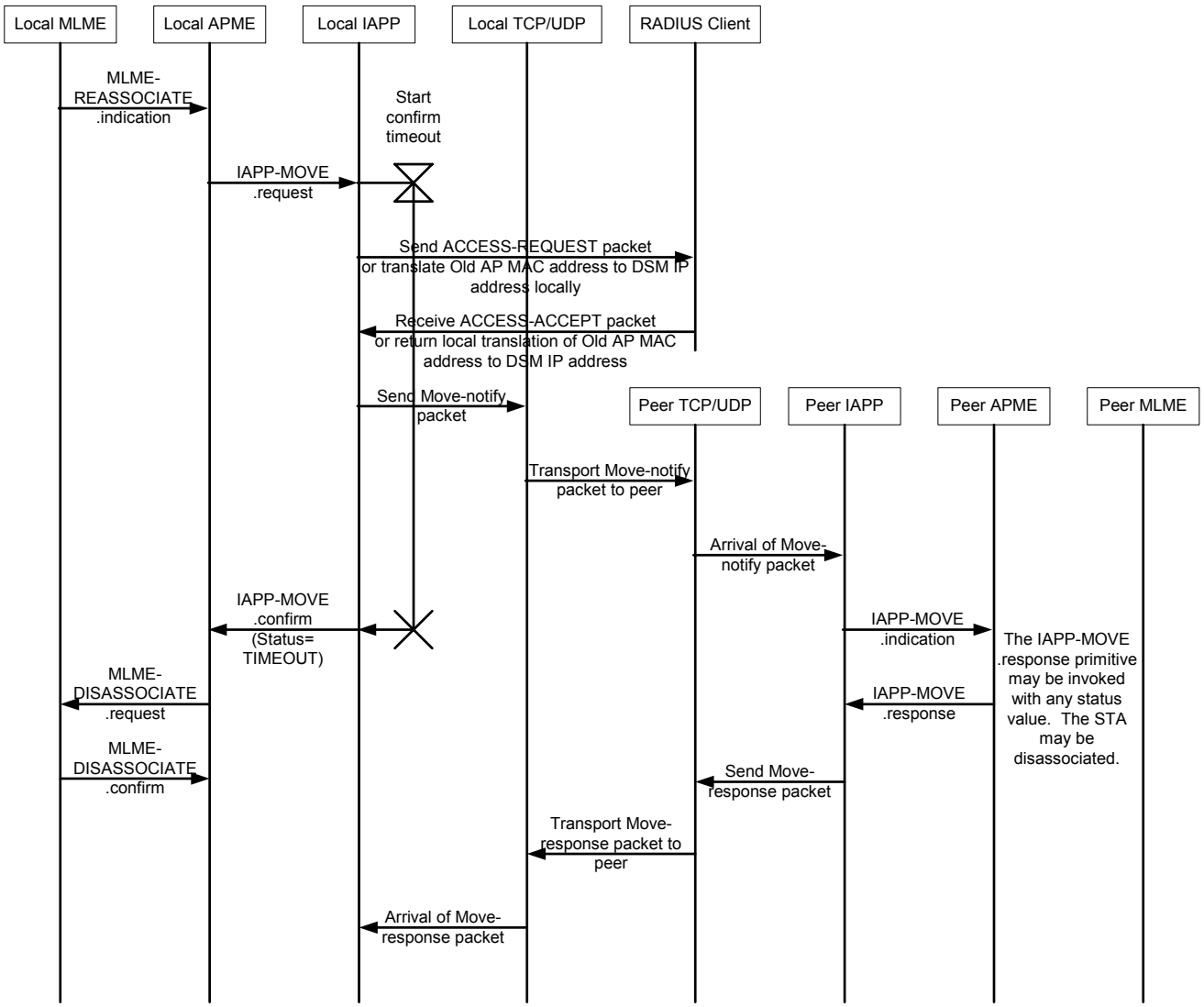

Figure 18, STA reassociation - timeout

5 Operation of the IAPP

The IAPP is a communication protocol, used by the management entity of an AP to communicate with other APs, when various local events occur in the AP. It is a part of a communication system comprising APs, STAs, an arbitrarily connected DS, and RADIUS infrastructure containing zero or more RADIUS servers. The RADIUS servers provide two functions, mapping the BSSID of an AP to its IP address on the DSM and distribution of keys to the APs to allow the encryption of the communications between the APs. The function of mapping the BSSID of an AP to its IP address on the DSM can also be accomplished by local configuration information or the IETF Inverse Address Resolution Protocol (RFC 2390). The function of the IAPP is to facilitate the creation and maintenance of the ESS, support the mobility of STAs, and enable APs to enforce the requirement of a single association for each STA at a given time, as stated in ISO/IEC 8802-11:1999.

5.1 IAPP Protocol Overview

IAPP supports three protocol sequences. One is initiated by invoking the IAPP-ADD.request after the APME receives an MLME-ASSOCIATE.indication.The second is initiated by invoking the IAPP-MOVE.request after the APME receives an

MLME-REASSOCIATE.indication. The third is initiated by invoking the IAPP-CACHE.request to cache context in neighboring APs to facilitate fast roaming.

5.1.1 Actions triggered by the IAPP-ADD.request

When the IAPP receives an IAPP-ADD.request it should send an IAPP ADD-notify packet and a Layer 2 Update Frame. The IAPP ADD-notify packet is an IP packet with a destination-IP-address of the IAPP IP multicast address, the source IP address, and MAC address of the AP. The message body contains the MAC address of the STA and the Sequence Number from the Association request sent by the STA. On receiving this message the APME should check its association table and remove an association with the STA if it exists and is determined to be older than the association indicated by the ADD-notify packet. Note that purpose of the IAPP ADD-notify packet is to remove stale associations, not to modify the learning table. The learning table update is done by the Layer 2 Update frame (see sec.6.3). This frame has the source MAC address of the associating STA. This frame is used by receiving APs and other layer 2 devices to update their learning table.

5.1.2 Actions triggered by an IAPP-MOVE.request

When the IAPP receives an IAPP-MOVE.request it should send an IAPP MOVE-Notify packet to the old AP and get back a MOVE-response from the old AP. The IAPP MOVE-Response carries the Context block for the STA's association from the old AP to the new AP

The IAPP MOVE-Notify and MOVE-Response are IP packets carried in a TCP session between APs. The IP address of the old AP must be found by mapping the BSSID from the reassociate message to its IP address. This mapping is done using a RADIUS exchange or locally configured information mapping the BSSID of APs to their IP addresses on the DSM. For this exchange any standard RADIUS server that supports the Call Check service-type should work.

If it is desired to encrypt the IAPP MOVE-response packet, the RADIUS Reply to the new AP will include, in addition to the IP address of the old AP, reply items with Security Blocks for both the new and old APs. The Security Blocks each contain information for securing the AP-AP connection. This information is dynamically generated by the RADIUS server as the Security Blocks are constructed. The Security Blocks are encrypted using the APs' BSSID user password (see 5.3.7.2 and 5.3.7.3) in the RADIUS registry. The RADIUS server would have to have an add-on to create the Security Block.

The new AP sends the Security Block for the old AP, which it received from the RADIUS Server, as a Send-Security-Block packet. This is the first message in the IAPP TCP exchange between the APs. The old AP returns ACK-Security-Block packet. At this point both APs have the information to encrypt all further packets for this exchange between the APs.

5.1.3 Actions triggered by the IAPP-CACHE-NOTIFY.request

When the IAPP entity receives an IAPP-CACHE-NOTIFY.request from its APME, the IAPP entity should send the given context to each of the APs in its neighboring AP graph. This context is sent from the original IAPP entity to each neighboring AP in CACHE-notify packets. When each neighboring AP receives a CACHE-notify packet, its IAPP entity issues an IAPP-CACHE-NOTIFY.indication to its APME, which causes the APME to update the STA context cache by adding or updating the STA context entry corresponding to the IAPP-CACHE-NOTIFY.indication. After the neighboring APME has updated its cache, it issues an IAPP-CACHE-NOTIFY.response to its IAPP entity, which sends a CACHE-response packet back to the original AP. When the original AP's IAPP entity receives the CACHE-response packet, the original IAPP entity issues an IAPP-CACHE-NOTIFY.confirm to the original APME.

If CACHE-response packets are not received from each of the neighboring APs before the IAPP-CACHE-NOTIFY.request{RequestTimeout} expires, the IAPP entity should delete the neighboring APs that did not respond before the expiration of the RequestTimeout from the neighbor graph. Only when all neighboring APs fail to respond before the expiration of the RequestTimeout will the original IAPP entity issue an IAPP-CACHE-NOTIFY.confirm{Status=TIMEOUT} to the original APME. If any CACHE-response packets are received with a status of STALE_CACHE, the IAPP should issue an IAPP-CACHE-NOTIFY.confirm {Status=STALE_CACHE} and the APME should delete the corresponding station context entry from the local cache.

Security for this message should be provided as for the IAPP-MOVE.request, described above in section 5.1.2.

5.2 Formation and maintenance of the ESS

An ESS is a set of Basic Service Sets (BSSs) that form a single LAN, allowing an STA to move transparently from one BSS to another throughout the ESS. As described in ISO/IEC 8802-11:1999 the initialization of the first AP via the MLME-START.request(BSSType=Infrastructure) establishes the formation of an ESS. Subsequent APs that are interconnected by a common DS and that are started with the same SSID extend the ESS created by the first. IAPP is defined to be able to provide a secure handoff mechanism of STA information between APs in the same ESS. IAPP can use a central RADIUS registry to define AP members of an ESS. Three levels of support for ESS formation are possible with the IAPP capabilities described here: 1) no administrative or security support; 2) support for dynamic mapping of BSSID to IP addresses; and 3) support for encryption and authentication of IAPP messages. Level one support can be achieved by configuring each AP in the ESS with the BSSID to IP address mapping for all other APs in the ESS. This may be acceptable for a small ESS. Many ESS providers will need levels 2 or 3, which require RADIUS support. The remainder of this section describes requirements for support of levels 2 and 3.

To include RADIUS support, the RADIUS server and the AP RADIUS client must be configured with the shared secret and with each other's IP address. This must be done prior to the first AP in an ESS becoming operational. Each AP acting as a RADIUS client should have its own shared secret with the RADIUS server, different from that of any other AP, containing the damage caused by the compromise of the key at any single AP to only the compromised AP.

Since the roaming STA sends an 802.11 reassociation request frame to the new AP containing the BSSID it is roaming from, each RADIUS server must also be configured with the following information for each BSSID. From an IAPP point of view, this set of BSSID entries defines the members of an ESS.

a) BSSID,

b) RADIUS BSSID Secret at least 160 bits in length

c) IP address or DNS name, and

d) Cipher suites supported by the AP for the protection of IAPP communications.

If an APME is going to use the services of IAPP, additional steps, internal to the AP, are necessary. Before the issuance of the MLME-START.request(BSSType=Infrastructure), the APME should issue the IAPP-INITIATE.request.

The IAPP entity is invoked by the APME to initiate STA context transfer between the old AP and the new AP. The IAPP may invoke RADIUS to obtain mapping of the old BSSID to the DSM IP address of the old AP and the security information with which to secure the communications with the peer IAPP entity.

5.3 RADIUS Protocol Usage

For the IAPP entity to function correctly, it must have the ability to discover the DSM IP address of the old BSSID in the ESS using the old BSSID as a lookup key. To implement this capability, the use of the RADIUS Protocol (IETF RFCs 2865 and 2869) is recommended. RADIUS is also used to obtain the security information to secure the communication between IAPP entities. This address mapping and security information may be preloaded or cached.

5.3.1 RADIUS Registration Access-Request

Upon receipt of an IAPP-INITIATE.request primitive, the AP

a) should register as a valid member of the ESS, and

b) may establish a secure channel for broadcast communications to all APs in the ESS.

To register the AP's membership in the ESS, and to obtain the security parameters necessary for establishing a secure broadcast connection with all the other APs in the ESS, the AP sends a RADIUS Registration Access-Request packet to the RADIUS server with a Service-Type of IAPP-Register. The Registration Access-Request packet uses the AP's BSSID as the User-Name, the AP's BSSID Secret as the User-Password, and contains the global SSID as a vendor-specific attribute. This enables the RADIUS server to register the BSSID as a part of the ESS, and also to store the AP's BSSID Secret. The Registration Access-Request also contains the list of the AP's supported ESP transforms and ESP authentication algorithms, which allows the RADIUS server to determine the appropriate common supported cipher suite(s) to use for the ADD-Notify and MOVE-Notify packets. Without registering with the RADIUS server, the AP will not be able to make use of the address resolution function to determine the DSM IP address from the BSSID of an AP.

The RADIUS Registration Access-Request contains the following attributes:

Table 1 - RADIUS Registration Access-Request Attributes

Attribute Number		
1	User-Name	BSSID. The BSSID should be represented in ASCII format, with octet values separated by a "-". Example: "00-10-A4-23-19-C0".
2	User-Password	BSSID Secret, determined by the AP
4	NAS-IP-Address	AP's IP Address
6	Service-Type	IAPP-Register (value = 15)
26	Vendor-Specific	The following IEEE 802.11 vendor-specific attributes:
26-13277-4	SSID	The ASCII text SSID which denotes the ESS in which the BSSID is registering
26-13277-5	Supported-ESP-Authentication-Algorithms	The list of ISAKMP ESP Authentication IDs corresponding to the ESP Authentication algorithms supported by this AP (see Table 12)
26-13277-6	Supported-ESP-Transforms	The list of ISAKMP ESP Transform IDs corresponding to the ESP transforms supported by this AP (See Table 11)
32	NAS-Identifier (optional)	AP's NAS Identifier
80	Message-Authenticator	The RADIUS message's authenticator

Per RFC 2865, other RADIUS attributes may be included in the Registration Access-Request packet in addition to the ones listed above.

5.3.2 RADIUS Registration Access-Accept

Upon receipt of a Registration Access-Request from the AP, the RADIUS Server verifies that the AP is a valid member of the ESS. If the RADIUS Server permits the AP entrance into the ESS, it returns a Registration Access-Accept packet. Receipt of a valid RADIUS Registration Access-Accept packet both confirms that the AP is a valid member of the ESS, and also provides the AP with the appropriate security information for establishing a secure group communications channel for IAPP. For key rollover purposes, the parameters obtained by the AP from the RADIUS Registration Access-Accept should be cached for use in sending ADD-Notify packets.

When the RADIUS server responds with a Registration Access-Accept, the packet should contain the following attributes:

Table 2 - RADIUS Registration Access-Accept Attributes

Attribute Number	Attribute Name	Value
1	User-Name	BSSID
6	Service-Type	IAPP-Register (value = 15)
26	Vendor-Specific	The following IEEE 802.11 vendor-specific attributes (optional):
26-13277-7	ESS-New-ESP-Transform-Key	The ESP Transform key used to encrypt ADD-Notify packets when sending
26-13277-8	ESS-New-ESP-Authentication-Key	The ESP Authentication key used to authenticate ADD-Notify packets when sending
26-13277-9	ESS-Old-ESP-Transform-Key	The ESP Transform key that can be used to decrypt ADD-Notify packets when receiving, if the New-ESP-Transform-Key does not work
26-13277-10	ESS-Old-ESP-Authentication-Key	The ESP Authentication key that can be used to authenticate ADD-Notify packets when receiving, if the New-ESP-Authentication-Key does not work
26-13277-11	ESS-ESP-Transform-ID	ESP Transform ID of the algorithm to use when encrypting/decrypting ADD-Notify packets
26-13277-12	ESS-ESP-Authentication-ID	ESP Authentication ID of the algorithm to use when encrypting/decrypting ADD-Notify packets

26-13277-13	ESS-ESP-SPI	SPI used to identify ESP group SA
27	Session-Timeout	Number of seconds until the AP should reissue the Registration Access-Request packet to the RADIUS Server to obtain new keying information
80	Message-Authenticator	The RADIUS message's authenticator

The ESS-New-ESP-Transform-Key, ESS-New-ESP-Authentication-Key, ESS-Old-ESP-Transform-Key, and ESS-Old-ESP-Authentication-Key attributes are encrypted as described for the MS-MPPE-Send-Key attribute in RFC 2548.

Per RFC 2865, other RADIUS attributes may be included in the Registration Access-Accept packet in addition to the ones listed above.

5.3.3 RADIUS Registration Access-Reject

As described in 5.3.2, upon receipt of a Registration Access-Request from the AP, the RADIUS Server will verify that the AP is a valid member of the ESS. A Registration Access-Reject may be issued due to an AP not supporting the ESP Transform or ESP Authentication algorithm selected for use in securing the ADD-Notify, or for other RADIUS configuration reasons not discussed here.

If the RADIUS Server determines that the AP is not a valid member of the ESS, the RADIUS Server will respond to the AP's Registration Access-Request packet with an RADIUS Registration Access-Reject. The RADIUS Registration Access-Reject packet instructs the AP to issue IAPP-INITIATE.confirm (ResultCode= FAILURE).

5.3.4 RADIUS Access-Request

Upon receipt of an IAPP-MOVE.request primitive, the receiving AP

a) must establish that the Old BSSID is a valid member of the New BSSID's ESS, and

b) may establish a secure channel for communications with the Old BSSID

To verify the Old BSSID's identity, and also to obtain the security parameters necessary for establishing a secure connection with the Old BSSID, the New AP sends a RADIUS Access-Request packet to the RADIUS server. The RADIUS Access-Request packet is used to verify the identity of the Old AP, and to establish the communications parameters between the New AP and the Old AP. The parameters obtained in the RADIUS Access-Accept are used to communicate with the Old AP, and can be cached for use upon receipt of future IAPP-MOVE.request primitives. It is important to note that this RADIUS Access-Accept verifies the old BSSID, and does not authenticate the STA.

The RADIUS Access-Request contains the following attributes:

Table 3 - RADIUS Access-Request Attributes

Attribute Number		
1	User-Name	Old BSSID. The Old BSSID should be represented in ASCII format, with octet values separated by a "-". Example: "00-10-A4-23-19-C0".
2	User-Password	NULL
4	NAS-IP-Address (optional)	New AP's IP Address
6	Service-Type	IAPP-AP-Check (16)
26	Vendor-Specific	The following IEEE 802.11 vendor-specific attributes:
26-13277-1	IAPP-Liveliness-Nonce (optional)	A 32-byte nonce used to ensure liveliness of the secure IAPP traffic. This attribute should not be included if secure IAPP communications are not required by the AP.
30	Called-Station-Id	The WM MAC Address of the new BSSID with which the STA is reassociating, in ASCII format, with octet values separated by a "-". Example: "00-10-A4-23-19-C0". The SSID should be appended to the WM MAC address, separated from the MAC address with a ":". Example "00-10-A4-23-19-C0:Company WLAN".
32	NAS-Identifier (optional)	New BSSID's NAS Identifier
61	NAS-Port-Type	IAPP (25)
80	Message-Authenticator	The RADIUS message's authenticator

Per RFC 2865, other RADIUS attributes may be included in the Access-Request packet in addition to the ones listed above.

5.3.5 RADIUS Access-Accept

Upon receipt of an Access-Request from the New BSSID, the RADIUS Server will verify that the Old BSSID is a valid member of the ESS of which the New BSSID is a member. If the RADIUS Server determines that the Old AP and New AP should be able to communicate with each other via IAPP, the RADIUS Server will respond to the New AP's Access-Request packet with an Access-Accept packet. The RADIUS Access-Accept packet both confirms that the Old BSSID is a valid member of the ESS, and also provides both the Old and New AP with the appropriate security information for establishing a secure communications channel.

When the RADIUS server responds with Access-Accept, the Access-Accept packet should contain the following attributes:

Table 4 - RADIUS Access-Accept Attributes

Attribute Number	Attribute Name	Value
1	User-Name	Old BSSID
8	Framed-IP-Address	Old BSSID's IP Address
26	Vendor-Specific	The following IEEE 802.11 vendor-specific attributes:
26-13277-2	New-BSSID-Security-Block (optional)	Security Block encrypted using new BSSID's user-password, to be decrypted and used by the new BSSID
26-13277-3	Old-BSSID-Security-Block (optional)	Security Block encrypted using old BSSID's user-password, to be sent via IAPP from the new BSSID to the old BSSID, and decrypted and used by the old BSSID
80	Message-Authenticator	The RADIUS message's authenticator

Per RFC 2865, other RADIUS attributes may be included in the Access-Accept packet in addition to the ones listed above.

The New-BSSID-Security-Block VSA carries the security information needed by the new AP to decrypt and encrypt ESP packets. The New-BSSID-Security-Block is defined in 5.3.7.2

5.3.6 RADIUS Access-Reject

As described in 5.3.5, upon receipt of an Access-Request from the New AP, the RADIUS Server will verify that the Old BSSID is a valid member of the ESS. If the RADIUS Server determines that the Old BSSID and New AP should NOT be able to communicate with each other via IAPP, the RADIUS Server will respond to the AP's Access-Request packet with a RADIUS Access-Reject. The RADIUS Access-Reject packet instructs the New AP to issue an MLME-

REASSOCIATE.confirm(ResultCode= REFUSED) for the STA that caused the original MLME-REASSOCIATE.request primitive.

5.3.7 IAPP RADIUS vendor-specific attributes

Table 5 contains a list of the RADIUS Vendor-Specific Attributes (VSAs) used by the IAPP. The IEEE 802.11 vendor code is 13277.

Per RFC 2865, RADIUS Vendor-Specific Attributes should have the following form:

RADIUS Attribute Type (26)	Attribute Length	Vendor-ID (13277)	Vendor Type	Vendor Length	Attribute Data
Octets: 1	1	4	1	1	n

Figure 19 - RADIUS Vendor-Specific Attribute Format

Table 5 - IAPP RADIUS Vendor-Specific Attributes

Vendor Type	Attribute Name	Description
1	IAPP-Liveliness-Nonce	A 32-byte nonce used to ensure liveliness of the secure IAPP traffic. This attribute should not be included if secure IAPP communications are not required by the AP.
2	New-BSSID-Security-Block	Security Block encrypted using new BSSID's user-password, to be decrypted and used by the new BSSID
3	Old-BSSID-Security-Block	Security Block encrypted using old BSSID's user-password, to be sent via IAPP from the new BSSID to the old BSSID, and decrypted and used by the old BSSID
4	SSID	The ASCII text SSID which denotes the ESS in which the AP is registering its BSSID
5	Supported-ESP-Authentication-Algorithms	The list of ISAKMP ESP Authentication IDs corresponding to the ESP Authentication algorithms supported by this AP (see Table 12)
6	Supported-ESP-Transforms	The list of ISAKMP ESP Transform IDs corresponding to the ESP transforms supported by this AP (See Table 11)
7	ESS-New-ESP-Transform-Key	The ESP Transform key used to encrypt ADD-Notify packets when sending
8	ESS-New-ESP-Authentication-Key	The ESP Authentication key used to authenticate ADD-Notify packets when sending
9	ESS-Old-ESP-Transform-Key	The ESP Transform key that can be used to decrypt ADD-Notify packets when receiving, if the New-ESP-Transform-Key does not work
10	ESS-Old-ESP-Authentication-Key	The ESP Authentication key that can be used to authenticate ADD-Notify packets when receiving, if the New-ESP-Authentication-Key does not work
11	ESS-ESP-Transform-ID	ESP Transform ID of the algorithm to use when encrypting/decrypting ADD-Notify packets
12	ESS-ESP-Authentication-ID	ESP Authentication ID of the algorithm to use when encrypting/decrypting ADD-Notify packets
13	ESS-ESP-SPI	SPI used to identify ESP group SA.
14	New-BSSID-Security-Block-IV	A 4-byte nonce used as the initialization vector to encrypt and decrypt the New-BSSID-Security-Block attribute

5.3.7.1 IAPP-Liveliness-Nonce

The IAPP-Liveliness-Nonce VSA is a 32-byte nonce used to ensure liveliness of the secure IAPP traffic. This attribute should not be included if secure IAPP communications are not required by the AP.

5.3.7.2 New-BSSID-Security-Block

The New-BSSID-Security-Block is a Security Block encrypted using new BSSID's user-password, to be decrypted and used by the new BSSID. It is a variable length attribute that contains the security information from the RADIUS Server for the new AP. The content of the Security Block should be interpreted by the new AP, and should not be passed on to other APs.

The Security Block is a series of information elements. This block is encrypted with the new AP's RADIUS BSSID Secret, using the ESP Transform algorithm given to it in the ESS-ESP-Transform-ID attribute of the RADIUS Registration Access-Accept packet**.** The new AP authenticates this Security Block using the ESS-ESP-Authentication-ID algorithm, and decrypts it using the ESS-ESP-Transform-ID cipher (with New-BSSID-Security-Block-IV as IV) and its RADIUS BSSID Secret as the decryption key. The transform and authentication keys are derived from the RADIUS BSSID Secret by first expanding the secret by the following method.

secret1 = HMAC-SHA1(null, secret)
secret2 = HMAC-SHA1(null, secret || secret1)
secret3 = HMAC-SHA1(null, secret || secret2)
…
…
secretN = HMAC-SHA1(null, secret || secretN-1)

key = secret1 || secret2 || secret3 || … || secretN

The transform key is the first N bits and the authentication key is the next M bits (the values of N and M are dependent on the cipher suite). The new AP creates the SAs from the information in the Security Block and if it caches these SAs, uses the lifetime to remove the SAs from its cache. The format of the Information Element is shown in Figure 24. Information elements are defined to have a common general format consisting of a 2 octet Element ID field, a 2 octet length field, and a variable-length element-specific information field. Each element is assigned a unique Element ID as defined below. The Length field specifies the number of octets in the Information field.

Table 6 - Information Elements in the New-BSSID-Security-Block

Element ID	Length	Information
2	8	Security lifetime in seconds
3	32	ACK nonce
4	1	ESP transform number
5	1	ESP authentication number
6	4	SPI used to identify ESP SA to the old AP
7	Variable	key used by ESP Transform for ESP packets to the old AP
8	Variable	key used by ESP Authentication for ESP packets to the old AP
9	4	SPI used to identify ESP SA from the old AP
10	Variable	key used by ESP Transform for ESP packets from the old AP
11	Variable	key used by ESP Authentication for ESP packets from the old AP

5.3.7.3 Old-BSSID-Security-Block

The Old-BSSID-Security-Block is a Security Block encrypted using old BSSID's user-password, to be decrypted and used by the old AP. It is a variable length attribute that contains the security information from the RADIUS Server for the old AP. The content of the Security Block should not be interpreted by the new AP, but should be passed on to the old AP. The contents of the Old-BSSID-Security-Block attribute are defined in 6.6.

5.3.7.4 SSID

The SSID VSA is the ASCII text string SSID which denotes the ESS in which the AP is registering its BSSID. Since RADIUS VSAs have a separate length value, the SSID is not null-terminated.

5.3.7.5 Supported-ESP-Authentication-Algorithms

The Supported-ESP-Authentication-Algorithms VSA is a list of consecutive one-byte values that are ISAKMP ESP Authentication IDs corresponding to the ESP Authentication algorithms supported by this AP (see Table 12 for values).

5.3.7.6 Supported-ESP-Transforms

The Supported-ESP-Transforms VSA is a list of consecutive one-byte values that are ISAKMP ESP Transform IDs corresponding to the ESP Transformation algorithms supported by this AP (see Table 11 for values).

5.3.7.7 ESS-New-ESP-Transform-Key

The ESS-New-ESP-Transform-Key VSA contains the ESP Transform key used to encrypt and decrypt ADD-Notify packets transmitted and received by this AP. If a received ADD-Notify packet does not correctly decrypt using the ESS-New-ESP-Transform-Key, the ESS-Old-ESP-Transform-Key should be used to decrypt the ADD-Notify packet.

The contents of this VSA are encrypted as described for the MS-MPPE-Send-Key attribute in RFC 2548.

5.3.7.8 ESS-New-ESP-Authentication-Key

The ESS-New-ESP-Authentication-Key VSA contains the ESP Authentication key used to authenticate ADD-Notify packets transmitted and received by this AP. If a received ADD-Notify packet does not pass authentication using the ESS-New-ESP-Authentication-Key, the ESS-Old-ESP-Authentication-Key should be used to authenticate the ADD-Notify packet.

The contents of this VSA are encrypted as described for the MS-MPPE-Send-Key attribute in RFC 2548.

5.3.7.9 ESS-Old-ESP-Transform-Key

The ESS-Old-ESP-Transform-Key VSA contains the ESP Transform key used only to decrypt ADD-Notify packets received by this AP if the received ADD-Notify packet does not correctly decrypt using the ESS-New-ESP-Transform-Key. This key should never be used to encrypt ADD-Notify packets sent from this AP.

The contents of this VSA are encrypted as described for the MS-MPPE-Send-Key attribute in RFC 2548.

5.3.7.10 ESS-Old-ESP-Authentication-Key

The ESS-Old-ESP-Authentication-Key VSA contains the ESP Authentication key used only to authenticate ADD-Notify packets received by this AP if the received ADD-Notify packet does not pass authentication using the ESS-New-ESP-Authentication-Key. This key should never be used to authenticate ADD-Notify packets sent from this AP.

The contents of this VSA are encrypted as described for the MS-MPPE-Send-Key attribute in RFC 2548.

5.3.7.11 ESS-ESP-Transform-ID

The ESS-ESP-Transform-ID VSA is a one-byte attribute that denotes the ESP Transform algorithm chosen by the RADIUS server for encrypting and decrypting the ADD-Notify packets, using values selected from Table 11. The selected transform algorithm is also used to encrypt and decrypt the New-BSSID-Security-Block and Old-BSSID-Security-Block attributes sent in the RADIUS packets.

5.3.7.12 ESS-ESP-Authentication-ID

The ESS-ESP-Authentication-ID VSA is a one-byte attribute that denotes the ESP Authentication algorithm chosen by the RADIUS server for authenticating the ADD-Notify packets, using values selected from Table 12. The selected authentication algorithm is also used to authenticate the New-BSSID-Security-Block and Old-BSSID-Security-Block attributes sent in the RADIUS packets.

5.3.7.13 ESS-ESP-SPI

The ESS-ESP-SPI VSA is a 4-byte attribute that is the Security Parameter Index which is used by all members of the ESS to lookup the correct group SA for the ADD-Notify packet protection.

5.3.7.14 New-BSSID-Security-Block-IV

The New-BSSID-Security-Block-IV VSA is an 8-byte nonce used as the initialization vector to encrypt and decrypt the New-BSSID-Security-Block attribute.

5.4 Support for 802.11 context transfer

There are no requirements from the existing mechanisms of IEEE 802.11-1999 for the IAPP to carry context information between APs. However, should such mechanisms be defined that establish a requirement for the IAPP to carry context information between APs, that information will be carried in the Context Block of IAPP MOVE-notify and MOVE-response packets. The cryptographic protection of the information in the Context Block, should such protection be required, will be the responsibility of the standard defining the format of the information element carrying the authentication information.

5.5 AP to AP Interactions

5.5.1 Station Move Process

The interaction between APs in an ESS when a STA is added to the STAs associated with an AP as a result of an 802.11 reassociation request frame minimally comprises the exchange of the IAPP MOVE-notify and IAPP MOVE-response messages by the new AP at which the reassociation occurs and the old AP that formerly held the association of the STA, as well as the transmission of a Layer 2 Update frame by the new AP. See section 5.5.3 for the modified process when proactive caching is utilized. If security is needed for the IAPP MOVE-notify and IAPP MOVE-response packets, they are wrapped in ESP.

The purpose of exchanging the IAPP MOVE-notify and MOVE-response packets is to allow the new AP and old AP to exchange STA context information. An example of this STA context information is STA security information that may allow faster reauthentication of a STA on reassociation. The purpose of transmitting the Layer 2 Update frame is to cause any layer 2 devices, such as bridges and switches, to update any forwarding information they may hold regarding the STA identified by the MAC address in the SA field of the frame, so that frames destined for the STA are delivered to a point in the DS where the new AP can forward these frames into the BSS containing the STA.

The SPIs and keys for the Security Associations (SAs) for ESP are created by the RADIUS Server and sent to the new AP as the New-BSSID-Security-Block and Old-BSSID-Security-Block RADIUS Attributes. The new AP decrypts the New-BSSID-Security-Block using the configured cipher and its RADIUS BSSID Secret. The new AP creates the SAs from the information in the Security Block and if it caches these SAs, uses the lifetime to remove the SAs from its cache.

The new AP sends the old-BSSID-Security-Block to the old AP in the IAPP Send-Security-Block packet. The old AP authenticates and decrypts this Security Block using the configured cipher (with Date/Time as IV), HMAC-MD5, and its RADIUS BSSID Secret. The cipher and HMAC keys are derived from the RADIUS BSSID Secret by first expanding the secret by the following method.

secret1 = HMAC-SHA1(null, secret)
secret2 = HMAC-SHA1(null, secret || secret1)
secret3 = HMAC-SHA1(null, secret || secret2)

…
…
secretN = HMAC-SHA1(null, secret || secretN-1)

key = secret1 || secret2 || secret3 || … || secretN

Each secret is a 160-bit value, represented in big-endian format. The cipher key is the first N bits and the authentication key is the next M bits (the values of N and M are dependent on the cipher suite). The new AP creates the SAs from the information in the Security Block and if it caches these SAs, uses the lifetime to remove the SAs from its cache. The old AP checks the date/time stamp received against either its date/time, if there is no current SA, or the date/time stamp used to create the old SAs, if such SAs are present. If the stamp just received is greater, it removes the old SAs, and uses the new. If the stamps are the same, all the rest of the Security Block content is the same and can be silently discarded. If the stamp just received is less, this is an invalid reply and should be ignored. If the old AP uses its own date/time it should allow a minimum of 5 minutes as an acceptable difference between its own date/time and a received date/time stamp, before discarding SAs it has cached and replacing them with a newly received SA.

The old AP takes the New-AP-ACK-Authenticator and sends it to the new AP in the IAPP ACK-Security-Block packet. The new AP authenticates and decrypts the New-AP-ACK-Authenticator using the configured cipher (with Date/Time as IV), HMAC-MD5, and its RADIUS BSSID Secret. The same password expansion routine is used here. It compares the nonce in this block with the nonce it received in the New-BSSID-Security-Block. If they are the same, the old AP is ready to receive the IAPP MOVE-notify protected with ESP. If they do not match, there was some attack or failure. The new AP CAN wait to see if another IAPP ACK-Security-Block packet arrives with the proper nonce or the new AP can resend the IAPP Send-Security-Block packet.

5.5.2 Station Add Process

The interaction between APs in an ESS as a result of an AP receiving an 802.11 association request frame comprises the transmission by the AP at which the association occurs of an IAPP ADD-notify packet and the transmission of a Layer 2 Update frame. The IAPP ADD-notify packet is sent to the IAPP IP multicast address. The IAPP multicast address is 224.0.1.178. The Layer 2 Update frame is sent to the MAC broadcast address and uses the MAC address of the STA that has associated as the MAC source address for the frame. See clause 6.2 for further information on the IAPP ADD-notify packet and clause 6.3 for further information on the Layer 2 Update frame. See section 5.5.3 for the modified process when proactive caching is utilized.
The purpose of transmitting the IAPP ADD-notify packet is to provide an indication to an AP that may have held an older association of a STA that has more recently associated with another AP that the AP holding that older association may discard any context for that STA. This should allow for more efficient management of AP resources. The purpose of transmitting the Layer 2 Update frame is to cause any layer 2 devices, such as bridges and switches, to update any forwarding information they may hold regarding the STA identified by the MAC address in the SA field of the frame, so that frames destined for the STA are delivered to a point in the DS where the new AP can forward these frames into the BSS containing the STA.

There is no security provided for the Layer 2 Update frame. If security is needed for the IAPP ADD-notify packet, it is wrapped in ESP. The Layer 2 Update frame does not open new potentials for attacks against the WLAN or the STAs. However, the ADD-notify is a UDP IP frame that COULD be sent from anywhere in the DS and attack the AP's state for the STA.

The SPI and keys for the Security Association (SA) for ESP are created by the RADIUS Server and sent to the AP as the RADIUS Attributes. The AP creates the SA from the information in the RADIUS response, caches the SA, and uses the registration session timeout to remove the SA from its cache.

At any time, there could be two broadcast SPIs for the ESS, as lifetime expires on each AP and the AP performs a new RADIUS Registration Access-Request/Access-Accept interaction. ADD-Notify packets are always sent with the newest SA, but the old SA might be needed to decrypt a received ADD-Notify.

5.5.3 Station Cache Process

The interaction between STAs in an ESS when a STA associates or reassociates comprises the exchange of the IAPP-CACHE.request and IAPP-CACHE.response messages by the new and neighbor AP's respectively.

The purpose of exchanging the messages is to proactively cache the context of the STA at neighbor APs in anticipation of the STA roaming to one of these AP's.

The security for these messages is provided in the same fashion as for the IAPP-MOVE series of messages.

5.6 Proactive Caching

Proactive caching is a method that supports fast roaming by caching the context of a STA in the APs to which the STA may roam. The next AP's are identified dynamically, i.e. without management pre-configuration, by learning the identities of neighboring APs. This section details the learning algorithm, and the proactive cache algorithm.

5.6.1 Neighbor Graphs and Dynamic Learning

A neighbor graph is the set of neighbors relative to a given AP. This set is kept by an AP so that it's neighbors can be identified quickly. Rather than incur the management overhead of manually listing the neighbors for a given AP, the AP can learn its neighbors dynamically through the course of operation from information in REASSOCIATION-REQUEST frames, and IAPP-MOVE.Request frames. The AP can prevent the addition of bogus neighbors by adding only those APs where a RADIUS Access-Accept message is returned by the RADIUS server.

The exact form of the implementation of neighbor graphs is vendor dependent, but it is suggested that a least recently used (LRU) cache be used since some neighbors will be mis-identified due to STA moves without radio operation, e.g. when a laptop is closed. In these cases, the STA will fail to disassociate and then will reassociate at another AP which may or may not be a valid neighbor. Since these events will occur less frequently than hand-offs to valid neighbors, a LRU cache will, over time, flush the invalid entries from the neighbor graph. The added benefit is that the neighbor graph size can be fixed permitting easier memory management.

5.6.2 Proactive Cache Algorithm

This section provides pseudo-code for the proactive cache method for an AP.

```
        /* This function performs a cachelookup */
      Lookup(STA_MAC, Old_AP_MAC) {
        /* Key is STA_MAC and Old_AP_MAC */
        if (<STA_MAC,Old_AP_MAC> in cache) {
              return context;
        } else {
        return NULL;
        }
      }

      /* This function inserts an entry in the cache */
      Insert(STA_MAC, Old_AP_MAC, context) {
        /* Key for insert is STA_MAC */
        If (STA_MAC in cache) {
              /* cache line is <STA_MAC, Old_AP_MAC,context> */
              Replace cache line;
        } else {
              replace_oldest_entry(STA_MAC,Old_AP_MAC,context);
        }
        return;
}
      /* Main function called  to determine which operation to perform: IAPP-MOVE or IAPP-CACHE
         when proactive caching is being used */
```

```
Proactive_Cache(STA_MAC, Old_AP_MAC) {
 /* Cache hit */
 If (Lookup(STA_MAC, Old_AP_MAC)) {
       Send REASSOCIATE.response to STA
 } else {
 /* Cache miss */
       /* Inform old AP of move via normal IAPP. */
       Send IAPP-MOVE.request to Old_AP
       /* Context is from IAPP-MOVE.response */
       Insert(STA_MAC,Old_AP_MAC,context);
 }
 /* update neighbor cache */
 update_cache(Old_AP_MAC);

 /* Push context out to neighbors */
 For each AP in Neighbors {
       Send IAPP-CACHE-NOTIFY.request to AP;
 }
 return;
}
```

5.6.3 Consitency of the Cache

The consitency of the cache is context dependent and APME implementations should ensure that IAPP-CACHE-NOTIFY.request primitives are issued whenever a STA context has changed on the AP. IAPP-CACHE-NOTIFY.indicate messages received with a lower sequence number than previously received for a particular MAC address should be responded to with an IAPP-CACHE-NOTIFY.response{Status=STALE_CACHE}.

5.7 AP specific MIB

An SNMP MIB using SMIv2 for the IAPP is defined in Annex A. The MIB contains attributes for the IAPP that are useful in monitoring and diagnosis of the operation of the IAPP.

5.8 Single station association

IEEE 802.11 specifies that each STA may only be associated with a single AP at any given time. (See 802.11-1999 subclauses 5.4.2.2 and C.2) When a STA changes its association from one AP to another, the STA issues a reassociation request frame (as specified in the 802.11 standard). Reception of the reassociate frame and granting of the association by the new AP causes the APME in that AP to issue an IAPP-MOVE.request service primitive. This causes an IAPP MOVE-notify packet to be sent to the Old AP, requesting the old AP to remove the STA from its table, to forward any stored context for the STA, and the new AP to add the STA and context to its own table. Thus, the use of the reassociation request frame by the STA allows the APs to ensure that there is only a single association for the STA.

When a roaming STA associates with an AP, rather than reassociates, the AP attempts to enforce the single STA association requirement by sending an IAPP ADD-notify packet and the Layer 2 Update frame to the DS. Because this packet is addressed to the IAPP multicast address (see 6.2), this packet may not reach all APs in an ESS. In particular, if the ESS spans multiple subnets, neither the ADD-notify packet nor the Layer 2 Update frame is likely to reach the APs on subnets other than the one on which the transmissions originate. If the old AP receives the IAPP ADD-notify packet, it should remove any context stored for the STA.

6 Packet Formats

6.1 General IAPP Packet Format

The general format of an IAPP packet is shown in Figure 20. An IAPP packet is carried in the TCP or UDP protocols over IP. The port number assigned to IAPP is 3517[3].

IAPP Version	Command	Identifier	Length	Data
Octets: 1	1	2	2	0-n

Figure 20 - General IAPP Packet Format

6.1.1 IAPP Version Field

The IAPP Version Field indicates the protocol version of the IAPP, and thus the organization of the rest of the packet. The value of the Version field for this protocol is zero. All other values are reserved. A device that receives a packet with an IAPP Version level that it does not support should silently discard the packet.

6.1.2 Command Field

This is an 8-bit integer value that identifies the specific function of the packet. The data field that is specific to that command follows each command field.

Table 7 - Command field values

Value	Command
0	ADD-notify
1	MOVE-notify
2	MOVE-response
3	Send-Security-Block
4	ACK-Security-Block
5	CACHE-notify
6	CACHE-response
7-255	Reserved

6.1.3 Identifier Field

The two-octet Identifier field aids in matching requests and responses. When sending an IAPP request packet, the value of the Identifier field should be unique, with respect to other outstanding packets. When sending an IAPP response packet the value of the Identifier field will be a copy the value of the Identifier field from the received request packet. The Identifier field can be used to help detect duplicate requests and responses. Duplicate requests and responses should be silently discarded.

[3] Port numbers are assigned by IANA.

6.1.4 Length Field

The two-octet Length field indicates the length of the entire packet, including the version, command, identifier, length and data fields. Octets outside the range of the Length field should be treated as padding and ignored on reception. If the packet is shorter than the Length field indicates, it should be silently discarded.

6.1.5 Data Field

The Data Field is a variable length field, the content of which is dependent on the value of the Command field. The content of the Data Field is described in 6.2, 6.4, 6.5, 6.6, and 6.9. for each of the packet types.

6.2 ADD-notify Packet

The ADD-notify packet is sent, using the IAPP over UDP and IP, on the local LAN segment to notify any AP that receives it that the STA identified in the packet has associated at the AP sending the packet. The packet is sent to the IAPP IP multicast address (see RFC 1112), so that it will reach every device on the DSM local subnet, even if the LAN is switched. The IAPP Multicast address is 224.0.1.178.

The ADD-notify packet carries the MAC address and sequence number from the STA that has associated with the AP. The format of the packet data field is shown in Figure 21.

Address Length	Reserved	MAC Address	Sequence Number
Octets: 1	1	n = Address Length	2

Figure 21 - ADD-notify Data Field Format

The Address Length is an 8-bit integer that indicates the number of octets in the MAC Address. This field allows the extension of the IAPP to IEEE 64-bit MAC addresses, when those become generally deployed. The Reserved field is reserved in this version of the protocol and should be sent with a value of zero. The Reserved field should be ignored on reception. The length of the Reserved field is one octet, in order to align the MAC Address field on a 16-bit boundary. The MAC Address is the MAC address of the STA that has associated. The length of the MAC Address field is equal to the value of the Address Length field. The Sequence Number field contains the integer value of the sequence number of the association request frame received by the AP from the STA that has associated. Allowable values for the Sequence number are between 0 and 4095.

6.3 Layer 2 Update Frame

The Layer 2 Update frame is an 802.2 Type 1 Logical Link Control (LLC) Exchange Identifier (XID) Update response frame. This frame is sent using a MAC source address equal to the MAC address of the STA that has associated, so that any layer 2 devices, e.g., bridges, switches and other APs, can update their forwarding tables with the correct port to reach the new location of the STA. The format of an XID Update frame carried over 802.3 is shown in Figure 22. The 802.3 MAC header is shown as an example only. Other MAC protocols than 802.3 may be used.

MAC DA	MAC SA	Length	DSAP	SSAP	Control	XID Information Field
Octets: 6	6	2	1	1	1	3

Figure 22 - Layer 2 Update Frame Format

The MAC DA is the broadcast MAC address. The MAC SA is the MAC address of the STA that has just associated or reassociated. The Length field is the length of the information following this field, eight octets. The value of both the DSAP and SSAP is null. The Control field and XID Information field are defined in IEEE Standard 802.2.

6.4 MOVE-notify Packet

The MOVE-notify packet is sent using the IAPP, over TCP/IP. This packet is sent from the AP directly to the old AP with which the reassociating STA was previously associated. TCP is used, rather than UDP, because of its defined retransmission behavior and the need for the exchange to be reliable.

The data field of the MOVE-notify packet carries the MAC address and sequence number from the STA that has reassociated with the AP sending the packet. The format of the data field for this packet is shown in Figure 23.

Address Length	Reserved	MAC Address	Sequence Number	Length of Context Block	Context Block
Octets: 1	1	n = Address Length	2	2	m = Length of Context Block

Figure 23 - MOVE-notify Data Field Format

The Address Length is an 8-bit integer that indicates the number of octets in the MAC Address. The Reserved field is reserved in this version of the protocol and should be sent with a value of zero. The Reserved field should be ignored on reception. The MAC Address is the MAC address of the STA that has requested reassociation. The Sequence Number field contains the integer value of the sequence number of the reassociation request frame received by the AP from the STA that has requested reassociation. Allowable values for the Sequence number are between 0 and 4095. The Length of Context Block is a 16-bit integer that indicates the number of octets in the Context Block field. The Context Block is a variable length field that contains the context information being forwarded for the reassociated STA indicated by the MAC Address. The content of the Context Block should not be interpreted by the IAPP.

The Context Block is a container for information defined in other 802.11 standards that needs to be forwarded from one AP to another upon reassociation of a STA. The Context Block is a series of information elements. The format of the Information Element is shown in Figure 24. The element identifiers and format of the information element content are defined by the standards that use the IAPP to transfer context from one AP to another. Information elements are defined to have a common general format consisting of a 2 octet Element ID field, a 2 octet length field, and a variable-length element-specific information field. Each element is assigned a unique Element ID as defined in the standards that use the IAPP to transfer context between APs. The Length field specifies the number of octets in the Information field.

Users of the IAPP service should ignore information elements whose element identifier they do not understand, rather than discarding the entire IAPP MOVE-notify packet.

Element Identifier	Length	Information
Octets: 2	2	n = Length

Figure 24 - Information Element Format

6.5 MOVE-response Packet

The MOVE-response packet is sent using the IAPP, over TCP and IP. This packet is sent directly to the AP from which the MOVE-notify packet was received. TCP is used, rather than UDP, because of its defined retransmission behavior and the need for the exchange to be reliable.

The data field of the MOVE-response packet carries the MAC address of the reassociated STA and the context information pertaining to that STA. The format of the data field for this packet is shown in Figure 25.

Address Length	Status	MAC Address	Sequence Number	Length of Context Block	Context Block
Octets: 1	1	n = Address Length	2	2	m = Length of Context Block

Figure 25 - MOVE-response Data Field Format

The Address Length is an 8-bit integer that indicates the number of octets in the MAC Address. The Status field is an 8-bit integer that indicates the status resulting from the receipt of the MOVE-notify packet. The allowable values for the Status field are shown in Table 8. The values for the Status field are derived from the Status parameter of the IAPP-MOVE.response service primitive. The MAC Address is the MAC address of the STA that has reassociated. The Sequence Number field contains the integer value of the sequence number from the MOVE-notify packet that caused the generation of this packet. The Length of Context Block is a 16-bit integer that indicates the number of octets in the Context Block field. The Context Block is a variable length field that contains the context information being forwarded for the reassociated STA indicated by the MAC Address. The content of the Context Block should not be interpreted by the IAPP.

Table 8 - MOVE-response Status Values

Status Value	Definition
0	Successful
1	Move denied
2	Stale move
3-255	Reserved

6.6 CACHE-notify Packet

The format of the CACHE-notify packet is shown in Figure 26.

Address Length	Reserved	MAC Address	Sequence Number	Current AP	Context Length	Length of Context Block	Context Timeout
Octets: 1	1	n = Address Length	2	n	2	m = Length of Context Block	2

Figure 26 – CACHE-notify Data Field Format

The Address Length is an 8-bit integer that indicates the number of octets in the MAC Address. The Reserved field is reserved in this version of the protocol and should be sent with a value of zero. The Reserved field should be ignored on reception. The MAC Address is the MAC address of the STA that has requested reassociation. The Sequence Number field contains the integer value of the sequence number of the reassociation request frame received by the AP from the STA that has requested reassociation. Allowable values for the Sequence number are between 0 and 4095. The Current AP is the WM MAC address of the AP sending the CACHE-notify packet. The Length of Context Block is a 16-bit integer that indicates the number of octets in the Context Block field. The Context Block is a variable length field that contains the context information being forwarded for the reassociated STA indicated by the MAC Address. The content of the Context Block should not be interpreted by the IAPP.

The Context Block is a container for information defined in other 802.11 standards that needs to be forwarded from one AP to another upon reassociation of a STA. The Context Block is a series of information elements. The format of the Information Element is shown in Figure 24. The element identifiers and format of the information element content are defined by the standards that use the IAPP to transfer context from one AP to another. Information elements are defined to have a common

general format consisting of a 2 octet Element ID field, a 2 octet length field, and a variable-length element-specific information field. Each element is assigned a unique Element ID as defined in the standards that use the IAPP to transfer context between APs. The Length field specifies the number of octets in the Information field.

The Context Timeout is the number of seconds for which the neighboring AP should maintain the STA Context before removing it from the cache. Receipt of another CACHE-notify packet for this STA MAC Address should reset the Context Timeout timer.

6.7 CACHE-response Packet

The CACHE-response packet is sent using the IAPP, over TCP and IP. This packet is sent directly to the AP from which the CACHE-notify packet was received. TCP is used, rather than UDP, because of its defined retransmission behavior and the need for the exchange to be reliable.

The data field of the CACHE-response packet carries the MAC address of the reassociated STA and the corresponding sequence number. The format of the data field for this packet is shown in Figure 27.

Address Length	Status	MAC Address	Sequence Number
Octets: 1	1	n = Address Length	2

Figure 27 - CACHE-response Data Field Format

The Address Length is an 8-bit integer that indicates the number of octets in the MAC Address. The Status field is an 8-bit integer that indicates the status resulting from the receipt of the CACHE-notify packet. The allowable values for the Status field are shown in Table 8. The values for the Status field are derived from the Status parameter of the IAPP-CACHE-NOTIFY.response service primitive. The MAC Address is the MAC address of the STA that has reassociated. The Sequence Number field contains the integer value of the sequence number from the CACHE-notify packet that caused the generation of this packet.

Table 9 - CACHE-response Status Values

Status Value	Definition
0	Successful
1	Stale Cache
2-255	Reserved

6.8 Send-Security-Block packet

The Send-Security-Block packet is sent using the IAPP, over TCP and IP. This packet is sent from the AP directly to the old AP with which the reassociating STA was previously associated. TCP is used, rather than UDP, because of its defined retransmission behavior and the need for the exchange to be reliable.

The data field of the Send-Security-Block packet carries the security information needed by the old AP to decrypt and encrypt ESP packets. The format of the data field for this packet is shown in Figure 28.

Initialization Vector	Length of Security Block	Security Block
Octets: 8	2	m = Length of Security Block

Figure 28 - Send-Security-Block Data Field Format

The Initialization Vector is the first 8 bytes of the ACK nonce. The Length of Security Block is a 16-bit integer that indicates the number of octets in the Security Block field. The Security Block is a variable length field that contains the security information being forwarded from the RADIUS Server through the new AP to the old AP. The content of the Security Block should be interpreted by the IAPP.

The Security Block is a series of information elements. This block is encrypted with the old AP's RADIUS BSSID Secret, using the AP's configured cipher. The old AP has to authenticate and decrypt it first before processing it. The Authentication Block is a 16 byte field that contains the result of an HMAC-MD5 hash of the Security Block. The format of the Information Element is shown in Figure 24. Information elements are defined to have a common general format consisting of a 2 octet Element ID field, a 2 octet length field, and a variable-length element-specific information field. Each element is assigned a unique Element ID as defined below. The Length field specifies the number of octets in the Information field.

Table 10 - Information Elements in the Send-Security-Block Packet

Element ID	Length	Information
12	6 or 8	Old BSSID
1	8	Date/Time stamp
15	6 or 8	New BSSID
16	4 or 16	New BSSID IP address
2	8	Security lifetime in seconds
13	56	New-AP-ACK-Authenticator
4	1	ESP transform identifier
5	1	ESP authentication identifier
6	4	SPI used to identify ESP SA from new AP
7	Variable	key used by ESP Transform for ESP packets from the new AP
8	Variable	key used by ESP Authentication for ESP packets from the new AP
9	4	SPI used to identify ESP SA to the new AP
10	Variable	key used by ESP Transform for ESP packets to the new AP
11	Variable	key used by ESP Authentication for ESP packets to the new AP
14	16	HMAC authentication block

The ESP Transform and Authentication algorithms are defined by IANA at http://www.iana.org/assignments/isakmp-registry. The recommended minimum set of transforms is ESP_DES, as defined in RFC 2407. The recommended minimum set of authentication algorithms is HMAC_MD5 and HMAC_SHA. The values of the identifiers as of the last update of 2001 September 6 are shown in Table 11 and Table 12.

Table 11 - ESP Transform Identifiers

Transform Identifier	Value	Reference
RESERVED	0	[RFC2407]
ESP_DES_IV64	1	[RFC2407]
ESP_DES	2	[RFC2407]
ESP_3DES	3	[RFC2407]
ESP_RC5	4	[RFC2407]
ESP_IDEA	5	[RFC2407]
ESP_CAST	6	[RFC2407]
ESP_BLOWFISH	7	[RFC2407]
ESP_3IDEA	8	[RFC2407]
ESP_DES_IV32	9	[RFC2407]
ESP_RC4	10	[RFC2407]
ESP_NULL	11	[RFC2407]
ESP_AES	12	[Leech]
Reserved for private use	249-255	[RFC2407]

Table 12 - ESP Authentication Algorithm Identifiers

Transform Identifier	Value	Reference
RESERVED	0	[RFC2407]
HMAC-MD5	1	[RFC2407]
HMAC-SHA	2	[RFC2407]
DES-MAC	3	[RFC2407]
KPDK	4	[RFC2407]
HMAC-SHA2-256	5	[Leech]
HMAC-SHA2-384	6	[Leech]
HMAC-SHA2-512	7	[Leech]
HMAC-RIPEMD	8	[RFC2857]
RESERVED	9-61439	
Reserved for private use	61440-65535	

6.9 ACK-Security-Block packet

ACK-Security-Block packet is sent using the IAPP, over TCP and IP. This packet is sent from the old AP with which the reassociating STA was previously associated directly to the new AP. TCP is used, rather than UDP, because of its defined retransmission behavior and the need for the exchange to be reliable. The format of the data field for this packet is shown in Figure 29.

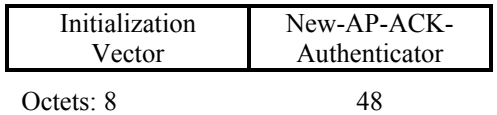

Figure 29 - ACK-Security-Block Data Field Format

The Initialization Vector is an 8-byte value copied from the Date/Time stamp. The New-AP-ACK-Authenticator field carries the content of the New-AP-ACK-Authenticator Information element that the old AP received in the Security Block. The content of the New-AP-ACK-Authenticator should be interpreted by the new AP. The New-AP-ACK-Authenticator is

encrypted with the new AP's RADIUS BSSID Secret, using the AP's configured cipher. The new AP has to authenticate and decrypt it first before processing it. This New-AP-ACK-Authenticator protects the new AP from spoofed ACK-Security-Block packets.

6.10 Information Element Definitions

The information elements defined in this recommended practice are listed in Table 13.

Table 13 - IAPP Information Elements

IAPP Element ID	Description
1	Date/Time stamp
2	Security lifetime
3	ACK nonce (32-byte)
4	ESP transform number
5	ESP authentication number
6	SPI from new AP
7	ESP transform key from new AP
8	ESP authentication key from new AP
9	SPI to new AP
10	ESP transform key to new AP
11	ESP authentication key to new AP
12	Old BSSID
13	New-AP-ACK-Authenticator (48-byte)
14	HMAC authentication block
15	New BSSID
16	New BSSID IP address
17 – 65,534	Reserved for future standardization
65,535	Proprietary Information. This information element must include the 3-byte Organizational Unique Identifier (OUI) from the organization's MAC address allocation as the first three bytes of the information field.

6.10.1 Date/Time stamp

The Date/Time stamp information element contains date and time information in RFC 1305 format. This information element is 8 octets in length.

6.10.2 Security lifetime

The Security lifetime information element contains a value indicating the seconds for the life of the SA. This value is used to compute the local time at which the SA is no longer valid for sending IAPP packets, and may be deleted. Common practice is to keep old SAs available for some limited time to receive packets from other APs.

6.10.3 ACK nonce (32-byte)

The ACK nonce information element is a 32 byte random value created by the RADIUS server, used by the new AP to establish liveness of the old AP. This information element is 4 octets in length.

6.10.4 ESP transform number

The ESP transform number information element is an 8-bit value that identifies the cryptographic algorithm used with ESP. This information element is 1 octet in length.

6.10.5 ESP authentication number

The ESP authentication number information element is an 8-bit value that identifies the authentication algorithm used with ESP. This information element is 1 octet in length.

6.10.6 SPI from new AP

The SPI from new AP information element is an Index into the SA for IAPP MOVE packets going from the new AP to the old AP. Initially this is a Request, but if this information is cached, it could later be a Response. This information element is 4 octets in length.

6.10.7 ESP transform key from new AP

The ESP transform key from new AP information element is a variable length value that is used by the cryptographic algorithm identified by the ESP transform number to encrypt information from the new AP to the old AP.

6.10.8 ESP authentication key from new AP

The ESP authentication key from new AP information element is a variable length value that is used by the authentication algorithm identified by the ESP authentication number to authenticate information from the new AP to the old AP.

6.10.9 SPI to new AP

The SPI to new AP information element is an Index into the SA for IAPP MOVE packets going to the new AP from the old AP. Initially this is a Response, but if this information is cached, it could later be a Request. This information element is 4 octets in length.

6.10.10 ESP transform key to new AP

The ESP transform key from new AP information element is a variable length value that is used by the cryptographic algorithm identified by the ESP transform number to encrypt information to the new AP from the old AP.

6.10.11 ESP authentication key to new AP

The ESP authentication key from new AP information element is a variable length value that is used by the authentication algorithm identified by the ESP authentication number to authenticate information to the new AP from the old AP.

6.10.12 Old BSSID

The Old BSSID information element contains the value of the BSSID for the old AP. This information element is variable length, either 6 or 8 octets.

6.10.13 New-AP-ACK-Authenticator (48-byte)

The New-AP-ACK-Authenticator information element contains a date/time stamp, an ACK nonce and an HMAC authentication block.

Table 14 - Content of the New-AP-ACK-Authenticator

Length	Information
8	Date/Time stamp
32	ACK nonce
16	HMAC authentication block

6.10.14 HMAC authentication block

The HMAC authentication block information element is a value created by performing an HMAC-MD5 operation over other designated fields. The usage of this information element is described in the definition of the particular packets that are authenticated. This information element is 16 octets in length.

6.10.15 New BSSID

The New BSSID information element contains the value of the BSSID for the new AP. This information element is variable length, either 6 or 8 octets.

6.10.16 New BSSID IP address

The New BSSID IP address information element contains the IP address of the new AP. This information element is variable length, either 4 or 16 octets.

Annex A, IAPP Management Information Base

(Normative)

```
-- ************************************************************************
-- * IEEE 802.11f Inter-AP Protocol Management Information Base
-- ************************************************************************

IEEE802dot11f-MIB DEFINITIONS ::= BEGIN
    IMPORTS
        MODULE-IDENTITY, OBJECT-TYPE,
        NOTIFICATION-TYPE,Integer32, Counter32  FROM SNMPv2-SMI

        DisplayString , MacAddress, RowStatus,
        TruthValue                              FROM SNMPv2-TC

        MODULE-COMPLIANCE, OBJECT-GROUP,
        NOTIFICATION-GROUP                      FROM SNMPv2-CONF

        ifIndex                                 FROM RFC1213-MIB;

-- ************************************************************************
-- *  MODULE IDENTITY
-- ************************************************************************

    iso             OBJECT IDENTIFIER ::= { 1 }
    member-body     OBJECT IDENTIFIER ::= { iso 2 }
    us              OBJECT IDENTIFIER ::= { member-body 840 }
    ieee802dot11    OBJECT IDENTIFIER ::= { us 10036 }

    ieee802dot11f MODULE-IDENTITY
    LAST-UPDATED "1301030000Z"
    ORGANIZATION "IEEE 802.11"
    CONTACT-INFO
           "WG E-mail: stds-802-11@ieee.org

                Chair: Stuart J. Kerry
              Postal: Philips Semiconductors, Inc.
                      1109 McKay Drive
                      M/S 48A SJ
                      San Jose, CA 95130-1706  USA
                 Tel: +1 408 474 7356
                 Fax: +1 408 474 7247
              E-mail: stuart.kerry@philips.com

              Editor: Bob O'Hara
              Postal: Black Storm Networks, Inc.
                      110 Nortech Parkway
                      San Jose, CA 95134 USA
                 Tel: +1 408 941 0500
                 Fax: +1 810 277 4718
              E-mail: bob@bstormnetworks.com"

    DESCRIPTION
        "The MIB module for IEEE 802.11f IAPP entities.
        iso(1).member-body(2).us(840).ieee802dot11(10036).iapp(6)"
    ::= { ieee802dot11 6 }

-- ************************************************************************
-- *  Major sections
-- ************************************************************************
--  IAPP diagnostic attributes
    --  DEFINED AS “The iappdiagnostics object class provides the necessary
```

```
    --  support at an 802.11 AP to manage and diagnose the IAPP processes
    --  and protocol in the AP such that the AP may work cooperatively as
    --  a part of an IEEE 802.11 network.";

iappdiagnostics OBJECT IDENTIFIER ::= {ieee802dot11f 1}

iappAPTable OBJECT-TYPE
      SYNTAX     SEQUENCE OF IappAPTableEntry
      MAX-ACCESS not-accessible
      STATUS     current
      DESCRIPTION
            "The (conceptual) table listing the other APs with
             which the AP has communicated via IAPP."
      ::= { iappdiagnostics 1 }

iappAPTableEntry OBJECT-TYPE
      SYNTAX     IappDiagnosticTableEntry
      MAX-ACCESS not-accessible
      STATUS     current
      DESCRIPTION
            "An entry (conceptual row) representing another AP
             with which the AP has communicated via IAPP."
      INDEX      { iappDiagnosticTableIndex }
      ::= { iappDiagnosticTable 1 }

IappAPTableEntry ::= SEQUENCE {
      iappAPTableIndex                              Integer32,
      iappAPIPAddress                               IpAddress,
      iappAPMACAddress                              MacAddress,
      iappClientServerPortNumber                    Integer32,
      iappAPRoundTripTime                           TimeTicks,
      iappAPRTO                                     TimeTicks,
      iappMoveNotifySent                            Counter32,
      iappMoveNotifyRetransmissions                 Counter32,
      iappMoveNotifyReceived                        Counter32,
      iappMoveResponseSent                          Counter32,
      iappMoveResponseReceived                      Counter32,
      iappMoveNotifyMalformed                       Counter32,
      iappMoveNotifyUnAuthentic                     Counter32,
      iappMoveResponseMalformed                     Counter32,
      iappMoveResponseUnAuthentic                   Counter32,
      iappMoveNotifyBadService                      Counter32,
      iappMoveResponseBadService                    Counter32,
      iappMoveNotifyPendingRequests                 Gauge32,
      iappMoveResponsePendingResponses              Gauge32,
      iappMoveNotifyTimeouts                        Counter32,
      iappUnknownType                               Counter32,
      iappMoveNotifyPacketsDropped                  Counter32,
      iappMoveResponsePacketsDropped                Counter32
}

iappAPTableIndex OBJECT-TYPE
      SYNTAX     Integer32 (1..2147483647)
      MAX-ACCESS not-accessible
      STATUS     current
      DESCRIPTION
             "A number uniquely identifying each other AP
             with which this AP has communicated via IAPP."
      ::= { iappAPTableEntry 1 }

iappAPIPAddress OBJECT-TYPE
      SYNTAX     IpAddress
      MAX-ACCESS read-only
      STATUS     current
      DESCRIPTION
            "The IP address of the AP
             referred to in this table entry."
```

```
      ::= { iappAPTableEntry 2 }

iappAPMACAddress OBJECT-TYPE
      SYNTAX     MacAddress
      MAX-ACCESS read-only
      STATUS     current
      DESCRIPTION
            "The MAC address of the AP
             referred to in this table entry."
      ::= { iappAPTableEntry 3 }


iappClientServerPortNumber  OBJECT-TYPE
      SYNTAX Integer32 (0..65535)
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
            "The UDP port the AP is using to send
             to the other AP.  The default value of this port is 3517."
      ::= { iappAPTableEntry 4 }

iappAPRoundTripTime  OBJECT-TYPE
      SYNTAX TimeTicks
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
            "The time interval (in hundredths of a second) between
             the most recent Move-Notify sent by this AP and the
             Move-Response that matched it from the other AP."
      ::= { iappAPTableEntry 5 }

iappAPRTO  OBJECT-TYPE
      SYNTAX TimeTicks
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
            "The Round Trip Timeout (RTO) (in hundredths of a second)
             between this AP and the other AP."
      ::= { iappAPTableEntry 6 }

-- Request/Response statistics
--
-- TotalIncomingPackets = MoveNotifyReceived + MoveResponseReceived + UnknownTypes
--
-- TotalIncomingPackets - Malformed - Unauthentic -
-- UnknownTypes - PacketsDropped = Successfully received
--

iappMoveNotifySent OBJECT-TYPE
      SYNTAX Counter32
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
            "The number of Move-Notify packets sent to this AP.
             This does not include retransmissions."
      ::= { iappAPTableEntry 7 }

iappMoveNotifyRetransmissions OBJECT-TYPE
      SYNTAX Counter32
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
            "The number of Move-Notify packets
             retransmitted to this AP."
      ::= { iappAPTableEntry 8 }

iappMoveNotifyReceived OBJECT-TYPE
```

```
      SYNTAX Counter32
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
            "The number of Move-Notify packets
             (valid or invalid) received from this AP."
      ::= { iappAPTableEntry 9 }

iappMoveResponseSent OBJECT-TYPE
      SYNTAX Counter32
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
            "The number of Move-Response packets sent to this AP."
      ::= { iappAPTableEntry  10 }

iappMoveResponseReceived OBJECT-TYPE
      SYNTAX Counter32
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
            "The number of Move-Response packets
             (valid or invalid) received from this AP."
      ::= { iappAPTableEntry 11 }

iappMoveNotifyMalformed OBJECT-TYPE
      SYNTAX Counter32
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
            "The number of malformed Move-Notify
             packets received from this AP.
             Malformed packets include packets with
             an invalid length. Unauthenticated packets
             or unknown types are not
             included as malformed packets."
      ::= { iappAPTableEntry 12 }

iappMoveNotifyUnAuthentic OBJECT-TYPE
      SYNTAX Counter32
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
            "The number of Move-Notify packets
             failing authentication, received from this AP."
      ::= { iappAPTableEntry 13 }

iappMoveResponseMalformed OBJECT-TYPE
      SYNTAX Counter32
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
            "The number of malformed Move-Response
             packets received from this AP.
             Malformed packets include packets with
             an invalid length. Unauthenticated packets
             or unknown types are not
             included as malformed packets."
      ::= { iappAPTableEntry 14 }

iappMoveResponseUnAuthentic OBJECT-TYPE
      SYNTAX Counter32
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
            "The number of Move-Response packets
             failing authentication, received from this AP."
```

```
      ::= { iappAPTableEntry 15 }

iappMoveNotifyBadService OBJECT-TYPE
      SYNTAX Counter32
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
            "The number of Move-Notify packets
             received from this AP which could not be acted
             upon, due to inclusion of an unavailable service.
             Malformed or unauthentic packets are not included
             in this count."
      ::= { iappAPTableEntry 16 }

iappMoveResponseBadService OBJECT-TYPE
      SYNTAX Counter32
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
            "The number of Move-Response packets
             received from this AP which could not be acted
             upon, due to requesting an unavailable service.
             Malformed or unauthentic packets are not included
             in this count."
      ::= { iappAPTableEntry 17 }

iappMoveNotifyPendingRequests OBJECT-TYPE
      SYNTAX Gauge32
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
            "The number of Move-Notify packets
             destined for this AP that have not yet timed out
             or received a response. This variable is incremented
             when a Move-Notify is sent and decremented due to
             receipt of a Move-Response, a timeout or retransmission."
      ::= { iappAPTableEntry 18 }

iappMoveNotifyTimeouts OBJECT-TYPE
     SYNTAX Counter32
     MAX-ACCESS read-only
     STATUS current
     DESCRIPTION
            "The number of Move-Notify timeouts to this AP.
             After a timeout the AP may retry or
             give up. A retry is counted as a
             retransmit as well as a timeout."
      ::= { iappAPTableEntry 19 }

iappUnknownType OBJECT-TYPE
      SYNTAX Counter32
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
            "The number of IAPP packets of unknown type which
             were received from this AP."
      ::= { iappAPTableEntry 20 }


iappMoveNotifyPacketsDropped OBJECT-TYPE
      SYNTAX Counter32
      MAX-ACCESS read-only
      STATUS current
      DESCRIPTION
            "The number of Move-Notify packets received from
             this AP and dropped for some other reason.
             Malformed or unauthentic packets, or those
```

```
                requesting an unavailable service are not included
                in this count."
        ::= { iappAPTableEntry  21 }

iappMoveResponsePacketsDropped OBJECT-TYPE
        SYNTAX Counter32
        MAX-ACCESS read-only
        STATUS current
        DESCRIPTION
              "The number of Move-Response packets received from
               this AP and dropped for some other reason, such
               as arriving after the Timeout window has expired.
               Malformed or unauthentic packets, or those
               requesting an unavailable service are not included
               in this count."
        ::= { iappAPTableEntry  22 }


-- ****************************************************************
-- *   End of IAPP MIB
-- ****************************************************************
END
```